



Ruben Miguel Álvaro de Lima

Licenciado em Ciências de Engenharia

Electrotécnica e de Computadores

Wave4IOPT - Editor e Visualizador Web de Formas de Onda

**– aplicação a controladores digitais especificados com
modelos de Redes de Petri IOPT –**

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: Prof. Doutor Luís Filipe dos Santos Gomes,
Professor Associado, Faculdade de Ciências e
Tecnologia da Universidade Nova de Lisboa

Júri:

Presidente: Prof. Doutor João Francisco Alves Martins

Arguente: Prof. Doutora Anikó Katalin Horváth da Costa

Vogal: Prof. Doutor Luís Filipe dos Santos Gomes



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2015

Wave4IOPT - Editor e Visualizador Web de Formas de Onda - aplicação a controladores digitais especificados com modelos de Redes de Petri IOPT -

Copyright © Ruben Miguel Álvaro de Lima, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Aos meus amigos e família

Agradecimentos

A realização desta dissertação de mestrado contou com importantes apoios e incentivos sem os quais não se teria tornado uma realidade e aos quais estarei eternamente grato.

Ao Professor Doutor Luís Gomes, pela sua orientação, conselhos, disponibilidade, dedicação e atenção dispensada ao longo da realização deste trabalho.

Ao Professor Fernando Pereira pela disponibilidade para solucionar problemas e dúvidas que foram surgindo ao longo da realização deste trabalho.

Aos meus amigos que estiveram ao meu lado durante esta fase, pelo companheirismo, força e apoio em certos momentos difíceis.

Por último, dirijo um agradecimento especial aos meus pais, pelo seu apoio incondicional, incentivo e paciência demonstrados, durante todo este período.

Os trabalhos associados a esta dissertação beneficiaram de resultados obtidos no projeto Petri-Rig – Ambiente de desenvolvimento de sistemas embutidos baseado em redes de Petri, financiado pela FCT com a referência PTDC/EEI-AUT/2641/2012.

Resumo

Este trabalho tem o objectivo de criar um Editor e Visualizador Web de Formas de Onda para controladores digitais especificados com modelos Redes de Petri *Input-Output Place-Transition* (IOPT). Após uma análise das ferramentas existentes e constatando-se a inexistência de uma ferramenta adequada a essa função, desenvolveu-se uma ferramenta denominada Wave4IOPT, que permite a visualização das formas de onda de sinais e eventos de entrada e de saída ao longo do tempo. A ferramenta permite também a visualização dos resultados do histórico de uma simulação de uma Rede de Petri IOPT, proveniente do Simulador das IOPT-Tools.

Esta ferramenta incorpora funcionalidades de edição, modos de visualização e um módulo básico de identificação e correcção de erros dos valores das formas de onda.

O Wave4IOPT está disponível a partir de um *browser* e prevê-se que venha a estar integrado no ambiente de ferramentas IOPT-Tools. Esta ferramenta foi construída utilizando tecnologias Web como HTML, JavaScript, CSS, SVG e JSON.

Adicionalmente, o Wave4IOPT poderá também servir para a edição, visualização e análise de outros tipos de sinais digitais, desde que sejam preenchidos os requisitos da estrutura do ficheiro JSON que será lido pela ferramenta.

Palavras-chave: Formas de Onda, Controladores digitais, Redes de Petri, IOPT.

Abstract

The aim of this work is to create a Web-based Waveform Editor and Viewer for digital controllers specified with Input-Output Place-Transition Petri net (IOPT) models. After an analysis of the available existing tools, it was identified that a tool with the above functionalities was inexistent, hence a Wave4IOPT tool was developed, allowing the visualization of waveforms associated with input and output signals and events throughout time. The tool also allows visualization of the result of a Petri-Net IOPT simulation, coming from the IOPT-Tools Simulator.

This tool incorporates functionalities of editing, visualization modes and a basic module of identification and correction of errors considering the values of the waveforms.

The Wave4IOPT is available through a browser and will be integrated in the IOPT-Tools framework. This tool was developed using Web technologies such as HTML, JavaScript, CSS, SVG, and JSON.

Additionally, Wave4IOPT will also be able to be used for editing, visualizing and analyzing other types of digital signals, as long as they meet the requirements of a JSON file structure, which will be read by the tool.

Keywords: Waveform, Digital Controllers, Petri Net, IOPT.

Índice de Matérias

Resumo	iii
Abstract	v
Índice de Matérias	vii
Índice de Figuras	ix
Índice de Tabelas	xiii
Símbologia e Notações	xv
1. Introdução	1
1.1 Motivação	1
1.2 Análise de editores e visualizadores de formas de onda existentes	2
1.3 Objectivos	3
1.4 Estrutura da dissertação	4
2. Conceitos básicos e Tecnologias	7
2.1 Formalismos de modelação	7
2.1.1 UML	7
2.1.1.1 Casos de Uso	8
2.1.1.2 Diagramas de actividade	12
2.1.2 Redes de Petri	13
2.1.2.1 Estrutura de uma Rede de Petri	14
2.1.2.2 Modelação com redes de Petri	16
2.1.2.3 Classe IOPT	17
2.2 IOPT-Tools	20
2.3 PNML	21
2.4 JSON	22
2.5 SVG	25
3. A ferramenta desenvolvida: WaveEditor4IOPT	27
3.1 Descrição da ferramenta	27

3.1.1 Dados recebidos e enviados	29
3.2 Funcionalidades	32
3.2.1 Diagrama de Casos de Uso	47
3.2.2 Descrição dos Casos de uso	48
3.2.3 Diagramas de Actividade	86
4. Exemplos práticos	91
4.1 Parque de estacionamento com uma entrada e uma saída	91
4.2 Parque de estacionamento com duas entradas e uma saída	100
5. Conclusões	107
Referências Bibliográficas	109
Referências Electrónicas	110
Anexo I	111
I.1. Ficheiro JSON1In1Out_beforeSim.	111
I.2. Ficheiro JSON1In1Out_afterSim.	121
I.3. Ficheiro JSON2In1Out_beforeSim	132
I.4. Ficheiro JSON2In1Out_afterSim.	147

Índice de Figuras

Figura 1.1: Visualizador de formas de onda de sinais digitais.	3
Figura 2.1: Caso de uso.	8
Figura 2.2: Actor.	8
Figura 2.3: a) Notação gráfica extend; b) Exemplo da notação gráfica extend	9
Figura 2.4: a) Notação gráfica include; b) Exemplo da notação gráfica include.	10
Figura 2.5: Exemplo de generalização/particularidade entre actores ou casos de uso.	11
Figura 2.6: Exemplo do gráfico da RdP.	14
Figura 2.7: Exemplo de marcação da RdP da Figura 2.6.	15
Figura 2.8: Evolução da marcação da RdP da Figura 2.7.	16
Figura 2.9: Rede de Petri de sistema produtor-consumidor com armazém com capacidade igual a três.	17
Figura 2.10: Adaptado de L. Gomes, F. Moutinho, F. Pereira 2013 [9].	20
Figura 2.11: Aspecto visual da IOPT-Tools com as suas funcionalidades [16].	21
Figura 3.1: Arquitectura geral do sistema.	27
Figura 3.2: Estrutura de uma Forma de Onda.	30
Figura 3.3: Aspecto visual do Wave4IOPT no browser Google Chrome.	32
Figura 3.4: Aspecto visual do menu de opções lateral do Wave4IOPT no browser Google Chrome.	34
Figura 3.5: Aspecto visual do Wave4IOPT após pressionar do botão Choose Waveforms.	36
Figura 3.6: Aspecto visual do painel de topo do Wave4IOPT.	37

Figura 3.7: Menu individual da Forma de Onda.	37
Figura 3.8: Sub-menu do Modificar Forma de Onda.	39
Figura 3.9: Sub-menu dos modos de visualização das Formas de Onda.	40
Figura 3.10: Forma de Onda no modo Compacto Simples.	40
Figura 3.11: Forma de Onda no modo Compacto por Níveis.	41
Figura 3.12: Forma de Onda no modo Bit a Bit.	41
Figura 3.13: Menu Bases Numéricas.	42
Figura 3.14: Antes do Incrementar de uma Forma de Onda de um evento.	42
Figura 3.15: Depois do Incrementar de uma Forma de Onda de um evento.	43
Figura 3.16: Antes do Decrementar de uma Forma de Onda no modo Compacto por Níveis.	43
Figura 3.17: Depois do Decrementar de uma Forma de Onda no modo Compacto por Níveis.	43
Figura 3.18: Antes do Incrementar com CTRL de uma Forma de Onda no modo Compacto Simples.	44
Figura 3.19: Depois do Incrementar com CTRL de uma Forma de Onda no modo Compacto Simples.	44
Figura 3.20: Depois do Incrementar com CTRL + ALT de uma Forma de Onda de um evento.	45
Figura 3.21: Depois do Incrementar com CTRL + ALT de uma Forma de Onda de um sinal no modo Compacto Simples.	45
Figura 3.22: Selecção dos passos a serem copiados.	46
Figura 3.23: Aspecto visual após uma cópia dos valores.	46
Figura 3.24: Antes da operação colar.	46
Figura 3.25: Após conclusão da operação colar.	46
Figura 3.26: Diagrama de Casos de Uso do Wave4IOPT.	47

Figura 3.27: Diagrama de Actividade dos Casos de Uso Incrementar valores nos passos, Incrementar com CTRL e Incrementar com CTRL + ALT.	86
Figura 3.28: Diagrama de Actividade dos Casos de Uso Decrementar valores nos passos, Decrementar com CTRL e Decrementar com CTRL + ALT.	87
Figura 3.29: Diagrama de Actividade do Caso de Uso Copiar valores dos passos.	88
Figura 3.30: Diagrama de Actividade do Caso de Uso Colar valores dos passos.	89
Figura 4.1: RdP IOPT de um controlador de entradas e saídas de um parque de estacionamento com uma entrada e uma saída.	92
Figura 4.2: Representação no Wave4IOPT do estado inicial da RdP antes da simulação do parque de estacionamento com 1 entrada e 1 saída.	95
Figura 4.3: Aspecto geral no Wave4IOPT após a simulação do parque de estacionamento com 1 entrada e 1 saída.	98
Figura 4.4: Representação com zoom a 150% no Wave4IOPT da RdP após a simulação do parque de estacionamento com 1 entrada e 1 saída.	99
Figura 4.5: RdP IOPT de um controlador de entradas e saídas de um parque de estacionamento com 2 entradas e 1 saída.	100
Figura 4.6: Representação no Wave4IOPT do estado inicial da RdP antes da simulação do parque de estacionamento com 2 entradas e 1 saída.	102
Figura 4.7: Aspecto geral no Wave4IOPT após a simulação do parque de estacionamento com 2 entradas e 1 saída.	104
Figura 4.8: Representação com zoom a 125% no Wave4IOPT da RdP após a simulação do parque de estacionamento com 2 entradas e 1 saída.	105

Índice de Tabelas

Tabela 3.1 – Descrição de Caso de Uso Incrementar valores nos passos.	48
Tabela 3.2 – Descrição de Caso de Uso Incrementar com CTRL.	49
Tabela 3.3 – Descrição de Caso de Uso Incrementar com CTRL + ALT.	50
Tabela 3.4 - Descrição de Caso de Uso Decrementar valores nos passos.	52
Tabela 3.5 – Descrição de Caso de Uso Decrementar com CTRL.	53
Tabela 3.6 – Descrição de Caso de Uso Decrementar com CTRL + ALT.	56
Tabela 3.7 - Descrição de Caso de Uso Copiar valores dos passos.	58
Tabela 3.8 - Descrição de Caso de Uso Colar valores dos passos.	59
Tabela 3.9 – Descrição de Caso de Uso Alterar nome da Forma de Onda.	60
Tabela 3.10 – Descrição de Caso de Uso Corrigir erros das Formas de Onda.	61
Tabela 3.11 – Descrição de Caso de Uso Alterar valor no passo.	62
Tabela 3.12 - Descrição de Caso de Uso Alterar a Base Numérica de uma Forma de Onda.	63
Tabela 3.13 – Descrição de Caso de Uso Recuar uma Acção.	65
Tabela 3.14 – Descrição de Caso de Uso Avançar uma Acção.	66
Tabela 3.15 – Descrição de Caso de Uso Ampliar.	66

Tabela 3.16 – Descrição de Caso de Uso Reduzir.	67
Tabela 3.17 – Descrição de Caso de Uso Alterar cor do nome da Forma de Onda.	68
Tabela 3.18 – Descrição de Caso de Uso Mostrar Forma de Onda.	69
Tabela 3.19 – Descrição de Caso de Uso Esconder Forma de Onda.	70
Tabela 3.20 – Descrição de Caso de Uso Alterar modo de visualização da Forma de Onda.	72
Tabela 3.21 – Descrição de Caso de Uso Mover Forma de Onda uma posição para cima.	74
Tabela 3.22 – Descrição de Caso de Uso Mover Forma de Onda uma posição para baixo.	75
Tabela 3.23 – Descrição de Caso de Uso Apagar Forma de Onda.	76
Tabela 3.24 – Descrição de Caso de Uso Duplicar Forma de Onda.	77
Tabela 3.25 – Descrição de Caso de Uso Abrir ficheiro JSON.	78
Tabela 3.26 – Descrição de Caso de Uso Guardar como ficheiro JSON.	80
Tabela 3.27 – Descrição de Caso de Uso Alterar número de passos visíveis.	81
Tabela 3.28 – Descrição de Caso de Uso Esconder todos as Formas de Onda.	83
Tabela 3.29 – Descrição de Caso de Uso Visualizar nome e número do passo de uma Forma de Onda.	84
Tabela 3.30 – Descrição de Caso de Uso Modificar Forma de Onda Tudo a 1.	84

Simbologia e Notações

AJAX	Asynchronous JavaScript and XML
CSS	Cascading Style Sheets
GIF	Graphics Interchange Format
HTML	HyperText Markup Language
IOPT	Input-Output Place-Transition
JPEG	Joint Photographics Experts Group
JSON	JavaScript Object Notation
LED	Light Emitting Diode
Petri-Rig	Petri net based framework for embedded systems engineering
PNG	Portable Network Graphics
PNML	Petri Net Markup Language
RdP	Rede de Petri
SVG	Scalable Vector Graphics
UML	Unified Modeling Language
VCD	Value Change Dump
XML	Extensible Markup Language



Introdução

Neste capítulo pretende-se apresentar a motivação que levou à realização deste trabalho, bem como, os seus objectivos e sua estrutura. De igual modo, será descrito de forma breve o estado de arte actual no que toca a editores e visualizadores de formas de onda utilizados para modelar controladores digitais.

1.1 Motivação

Desde a sua criação, os controladores digitais assumiram um papel relevante na modernização da Indústria. Hoje em dia, estão presentes em múltiplos sectores do nosso quotidiano, dando resposta a um número crescente de requisitos, implicando, contudo, uma crescente complexidade dos sistemas, o que conduz a um grau de exigência igualmente maior na sua modelação. Assim, existe uma necessidade cada vez maior de existirem novas ferramentas associadas a metodologias de modelação dos sistemas, de forma a tornar o desenvolvimento desses sistemas menos complexo, mais rápido e com menos custos.

O projecto Petri-Rig (Ambiente de desenvolvimento de sistemas embutidos baseado em redes de Petri) vai de encontro à necessidade descrita anteriormente. O trabalho associado a esta dissertação insere-se neste projecto, tendo como título:

Wave4IOPT - Editor e Visualizador Web de Formas de Onda – aplicação a controladores digitais especificados com modelos de Redes de Petri IOPT.

As Redes de Petri (RdP) é um formalismo de modelação de fácil compreensão, com uma representação gráfica bastante simples e flexível e com grande nível de abstracção, permitindo modelar sistemas complexos.

Dada a falta de ferramentas, disponíveis a partir de um *browser*, de formas de onda para controladores digitais especificados com modelos RdP IOPT, o autor considera que esta dissertação será certamente uma mais-valia.

1.2 Análise de editores e visualizadores de formas de onda existentes

Este sub-capítulo tem o intuito de descrever, de forma breve, o estado de arte actual no que toca a editores e visualizadores de formas de onda utilizados para modelação de controladores digitais especificados com modelos RdP IOPT.

Uma pesquisa mais específica, no principal *site* sobre RdP, *World of Petri Nets* [14], nada revelou quanto a existência de algum visualizador de formas de onda dedicado às RdP.

Numa pesquisa não tão específica como a anterior, chegou-se à conclusão de que existem algumas aplicações *desktop* de visualizadores de formas de onda e de apenas um visualizador de ondas baseado em *Web*: o *EDA Playground Wave* (EPWave). Contudo, ao contrário do Wave4IOPT, nenhum deles é específico para RdP, muito menos para a classe IOPT das RdP, não sendo possível acrescentar sinais e eventos externos ao sistema.

O EPWave [15] é, tal como a ferramenta Wave4IOPT, acedível através de um *browser* e dedicado a sinais digitais. Em vez de um ficheiro *JavaScript Object Notation*

(JSON), recebe ficheiros no formato *Value Change Dump (VCD)*, um tipo de ficheiro gerado por um Simulador pertencente ao mesmo conjunto de ferramentas ao qual pertence o *EPWave*: o *EDA Playground*. Conclui-se também que apresenta menos funcionalidades de visualização, quando comparado com o Wave4IOPT e apresenta um aspecto gráfico mais orientado para o hardware.

Na figura seguinte, apresenta-se uma imagem do *EPWave*:

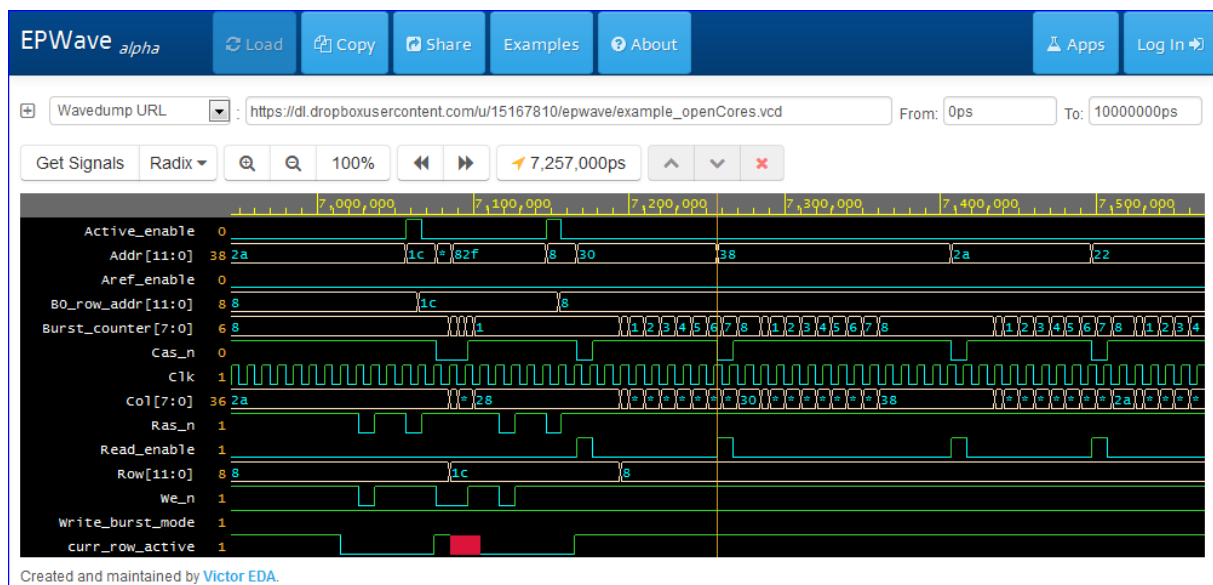


Figura 1.1: Visualizador de formas de onda de sinais digitais.

1.3 Objectivos

O objectivo é criar um Editor e Visualizador Web de Formas de Onda para controladores digitais especificados com modelos RdP IOPT.

Esta ferramenta estará acessível a partir de um *browser* e permitirá a visualização das formas de onda de sinais e eventos de entrada e de saída ao longo do tempo, resultado do histórico de uma simulação de uma RdP IOPT, proveniente do Simulador das IOPT-Tools.

Deste modo, pretende-se que a ferramenta receba um histórico de uma simulação do Simulador IOPT (via ficheiro JSON ou directamente do Simulador IOPT) de forma a que se possa visualizar toda a dinâmica da evolução da RdP IOPT

ao longo do tempo, tirando partido das vantagens da tecnologia *Scalable Vector Graphics* (SVG).

Para além disso, a ferramenta permitirá realizar diversas operações de edição a sinais e eventos de entrada e gravar essas alterações num ficheiro JSON aproveitando as vantagens desta tecnologia como ser de fácil leitura, leve e de ser utilizado por diversas aplicações Web. Possui também algumas funcionalidades de visualização das formas de onda, bem como, um módulo básico de identificação e correcção de erros dos valores das formas de onda.

Adicionalmente, a ferramenta poderá servir para editar, visualizar e analisar outros tipos de sinais digitais, desde que sejam preenchidos os requisitos da estrutura do ficheiro JSON que será lido pela ferramenta.

1.4 Estrutura da Dissertação

Esta dissertação foi estruturada em cinco capítulos, incluindo a introdução (primeiro capítulo).

É apresentada de seguida, uma descrição sucinta de cada um deles.

No segundo capítulo, apresentam-se e descrevem-se os fundamentos teóricos que fundamentam esta dissertação. Assim, apresentam-se alguns formalismos de modelação, como os Casos de Uso e Diagramas de Actividade do UML e Redes de Petri. No sub-capítulo dedicado às RdP, é abordado mais em pormenor aspectos da sua estrutura e modelação, bem como, conceitos da classe IOPT. Ainda no segundo capítulo são abordadas algumas tecnologias como as IOPT-Tools, PNML, JSON e SVG.

O terceiro capítulo é dedicado à ferramenta desenvolvida Wave4IOPT. Inicia-se com uma descrição da ferramenta, com algumas considerações ao nível da sua arquitectura e da estrutura dos dados enviados e recebidos. Seguidamente, são descritas todas as funcionalidades do Wave4IOPT. Primeiro de forma sucinta, acompanhada por algumas imagens. Depois, mais ao pormenor, com o auxílio dos

formalismos de modelação, como o diagrama de casos de uso, bem como, a descrição e modelo comportamental para cada caso de uso.

No quarto capítulo são apresentados exemplos práticos de aplicação da ferramenta Wave4IOPT desenvolvida, utilizando, nomeadamente, dois exemplos de Redes de Petri IOPT que modelam o controlador para um parque de estacionamento: com uma entrada e uma saída e outro com duas entradas e uma saída.

Finalmente, no quinto capítulo apresentam-se as conclusões da dissertação e sugestões para trabalhos futuros, de forma a darem continuidade a este trabalho.

Para além disso, esta dissertação inclui alguns anexos, sendo ficheiros utilizados no contexto do quarto capítulo.

De notar que se optou por escrever esta dissertação segundo o Acordo Ortográfico da Língua Portuguesa de 1990¹.

¹ em Diário da República n.º 193, Série I-A, Págs. 4370 a 4388 e Retificação n.º 19/91 de 7 de Novembro



Conceitos básicos e Tecnologias

Neste capítulo, pretende-se fazer uma breve apresentação de alguns Formalismos de Modelação utilizados nesta dissertação, bem como, conceitos teóricos de Redes de Petri e de algumas tecnologias utilizadas como o JSON e SVG.

2.1 Formalismos de Modelação

Este sub-capítulo serve o propósito de dar a conhecer alguns dos formalismos de modelação utilizados nesta dissertação.

2.1.1 UML

A *Unified Modeling Language* (UML) é uma linguagem para especificar, visualizar, contruir e documentar os artefactos dos sistemas de software, assim como, para modelação de negócios e outros sistemas que não software [1].

O UML é uma linguagem padrão para modelação amplamente adoptada internacionalmente pela indústria de engenharia de software e não só. Incorpora diagramas, notações gráficas do modelo do sistema, de forma a criar visualmente modelos fiéis ao sistema em análise.

2.1.1.1 Casos de Uso

Os Casos de Uso têm o propósito de identificar as funcionalidades requeridas para o sistema em análise, sendo utilizado na fase de levantamento e análise dos requisitos desse sistema.

O Diagrama de Casos de Uso é uma representação gráfica simples e geral de como se irá comportar o sistema a modelar. É constituído por diversas notações gráficas, identificando-se de seguida algumas que se irão adoptar :



Figura 2.1: Caso de uso

Actor



Figura 2.2: Actor

Extend – o comportamento do caso de uso estendido pode ser (não existe obrigatoriedade) estendido pelo comportamento de outro. O caso de uso estendido é definido independentemente do caso de uso que o estende, e faz sentido mesmo sem este (**Figura 2.3a**).

Tomando como exemplo um Cliente (Actor), o acto de pagar a refeição poderá ou não ser estendido pelo acto de dar gorjeta (**Figura 2.3b**).

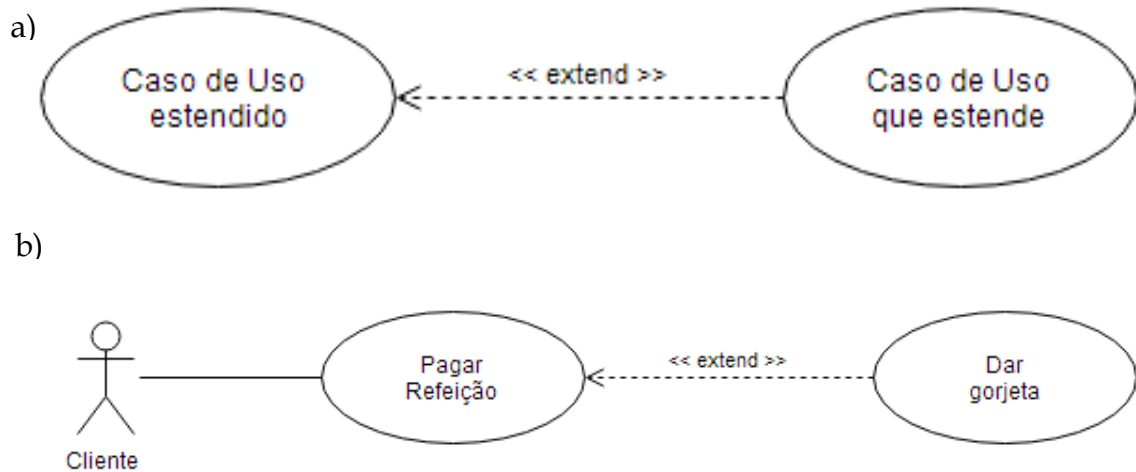


Figura 2.3: a) Notação gráfica extend; b) Exemplo da notação gráfica extend

Include – o caso de uso incluído faz parte do caso de uso que o inclui. O caso de uso incluído é obrigatório, sendo sempre necessário para a correcta execução do caso de uso que o inclui. O caso de uso incluído pode ser visto como uma função que é chamada/utilizada pelo caso de uso que o inclui. O sentido da seta transmite esta ideia, em que o que inclui depende do incluído, mas este não depende daquele que o inclui (**Figura 2.4a**).

No caso do exemplo concreto de um Condutor, ao fazer uma viagem, terá que incluir obrigatoriamente o abastecimento ao veículo (**Figura 2.4b**).

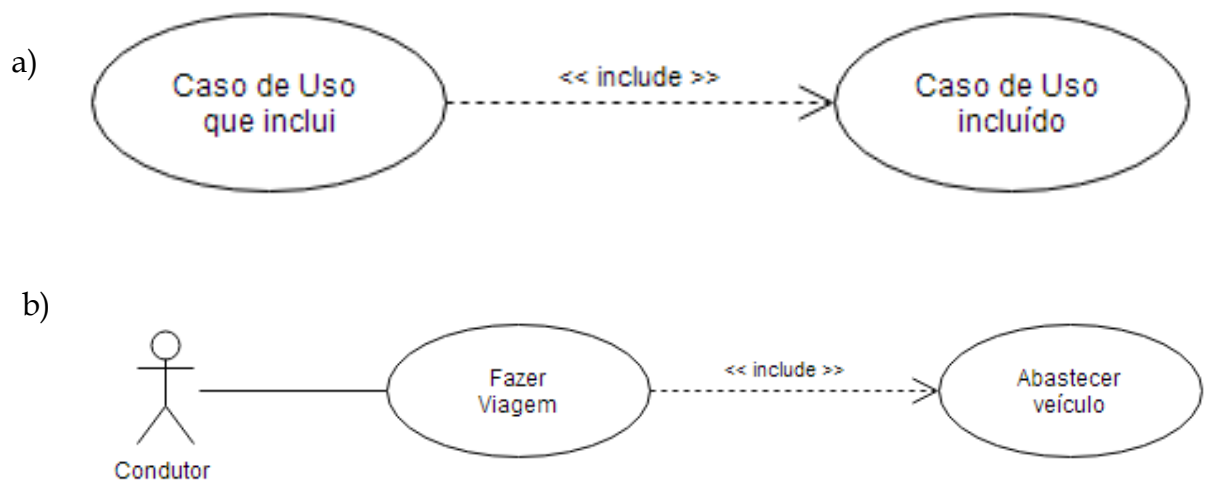


Figura 2.4: a) Notação gráfica include; b) Exemplo da notação gráfica include.

Generalização/particularidade entre actores ou casos de uso – remete para herança entre actores ou entre casos de uso.

No exemplo entre dois actores, o Estudante de Erasmus é um caso particular do Estudante. Note-se que o Estudante de Erasmus é um Estudante, pelo que faz tudo o que o Estudante faz.

No exemplo entre casos de uso, os casos de uso Efectuar Depósito e Efectuar Transferência são casos particulares do Efectuar Transacção (**Figura 2.5**). Deste modo, para se Efectuar Transferência terá que se executar o mesmo conjunto de acções do Efectuar Transacção, bem como adicionalmente, o conjunto de acções específicas do próprio caso de uso Efectuar Transferência. De igual modo, o Efectuar Depósito irá herdar o conjunto de acções do Efectuar Transacção, bem como, executar igualmente as acções específicas do caso de uso Efectuar Depósito.

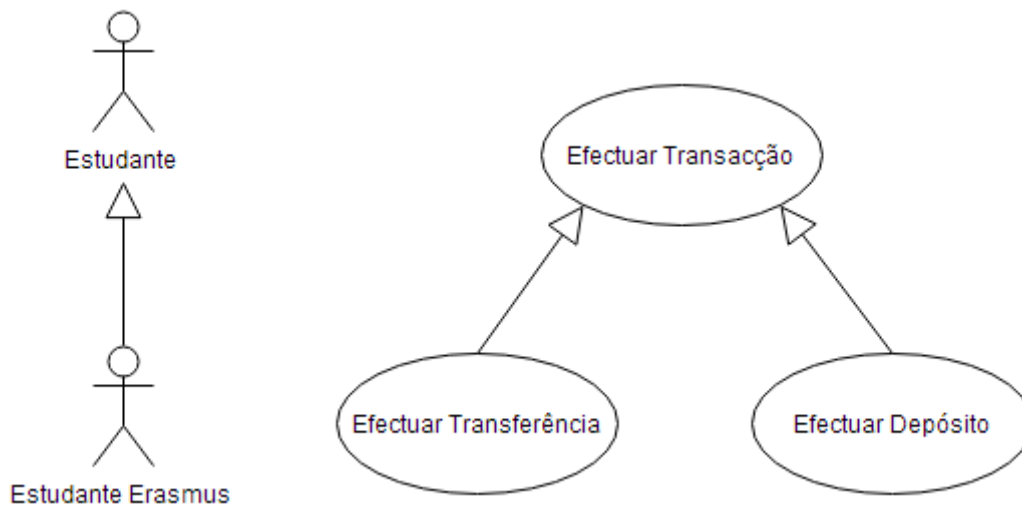


Figura 2.5: Exemplo de generalização/particularidade entre actores ou casos de uso.

Cada caso de uso é descrito de forma textual, utilizando-se uma determinada minuta (*template*) para o efeito, onde se detalha os diversos cenários possíveis inerentes ao sistema. A minuta poderá incorporar os seguintes tipos de informação [2]:

Nome ou Título do Caso de Uso – nome que transmite para que serve o caso de uso. Deve começar por um verbo.

Pré-condições – o que tem de ser verdade para que este caso de uso possa ser realizado.

Pós-condições – o que tem de ser verdade após a conclusão deste caso de uso.

Sumário – descrição muito resumida que indique os objectivos do caso de uso.

Actor – é alguém ou algo (programa ou dispositivo físico) que interage com o sistema executando o caso de uso, executando acções. Corresponde sempre a uma entidade que está fora do sistema em discussão.

Cenário principal – onde se especifica a sequência de acções pelo caminho principal, sem caminhos alternativos.

Cenários adicionais – onde se especificam os diversos cenários adicionais ou alternativos. São variantes dos cenário principal.

2.1.1.2 Diagramas de Actividade

Um Diagrama de Actividade UML é uma representação gráfica de um fluxo sequencial das actividades de um sistema. São usados para a construção do modelo comportamental do sistema. É especialmente útil para visualização de processos e fluxos de negócio ou casos de uso.

É constituído por diversas notações gráficas, tais como:



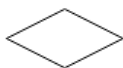
Nó inicial, em que marca o início da actividade.



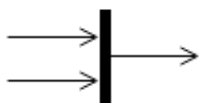
Nó final, em que marca o fim da actividade.



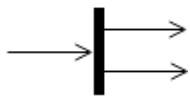
Acção, representando um único passo da actividade.



Nó de Decisão, onde são testadas condições. As condições poderão surgir representadas textualmente a acompanhar as transições de saída.



Nó de sincronização, onde fluxos paralelos ou concorrentes se juntam (do tipo *join*),



ou se separam (do tipo *fork*).



Transição, representado por um arco dirigido, situa-se entre quaisquer duas notações gráficas referidas anteriormente.

2.1.2 Redes de Petri

Em 1962, Carl Adam Petri [3], um matemático e cientista alemão, dava a conhecer ao mundo o conceito de Redes de Petri (RdP), através da sua tese de doutoramento “Comunicação com Autómatos”, na Universidade Técnica de Darmstadt. Na sua tese, Petri formulou as bases para a teoria da comunicação entre componentes assíncronos de um sistema computacional. Ele estava particularmente interessado na descrição das relações causais entre eventos. A sua tese, constituiu essencialmente um desenvolvimento teórico dos conceitos base a partir dos quais as Redes de Petri foram desenvolvidas [4].

As RdP são uma ferramenta de modelação matemática e gráfica aplicável a muitos sistemas. Sistemas caracterizados como sendo concorrentes, assíncronos, distribuídos, paralelos, não determinísticos, e/ou estocásticos.

Como ferramenta gráfica, as RdP podem ser usadas como um auxiliar de comunicação visual, similar a fluxogramas ou diagramas de blocos. Adicionalmente, marcas (*tokens*) são utilizadas nestas redes para simular a dinâmica e concorrência de actividades dos sistemas. Como ferramenta matemática, é possível configurar equações de estado, equações algébricas e outros modelos matemáticos que regem o comportamento de sistemas [5].

Com o passar dos anos, aumentou aplicabilidade das RdP, em muito devido às suas características de modelação, como a sincronização e concorrência entre processos e gestão de conflitos e partilha de recursos. Hoje em dia, as RdP estão presentes em diversas áreas da Engenharia, como a Engenharia de Software, de Comunicações, de Manufactura e Automação.

Assim, as RdP são um formalismo de modelação ideal para representar sistemas a eventos discretos, em que se recorre a uma caracterização do sistema em termos de estados observáveis no sistema e das transições possíveis entre esses estados [6].

Não se pretende realizar, nesta dissertação, uma introdução extensa e detalha às RdP. Deste modo, considerou-se a representação gráfica da estrutura de uma RdP como aspecto mais útil para ilustrar os seus conceitos teóricos [4].

2.1.2.1 Estrutura de uma Rede de Petri

Uma RdP é constituída por dois tipos de nós: lugares e transições. Os lugares são normalmente representados por círculos ou elipses e as transições por barras ou rectângulos. Os lugares e as transições são interligados por arcos dirigidos. Na **Figura 2.6**, está representado um exemplo demonstrativo da representação gráfica de uma RdP:

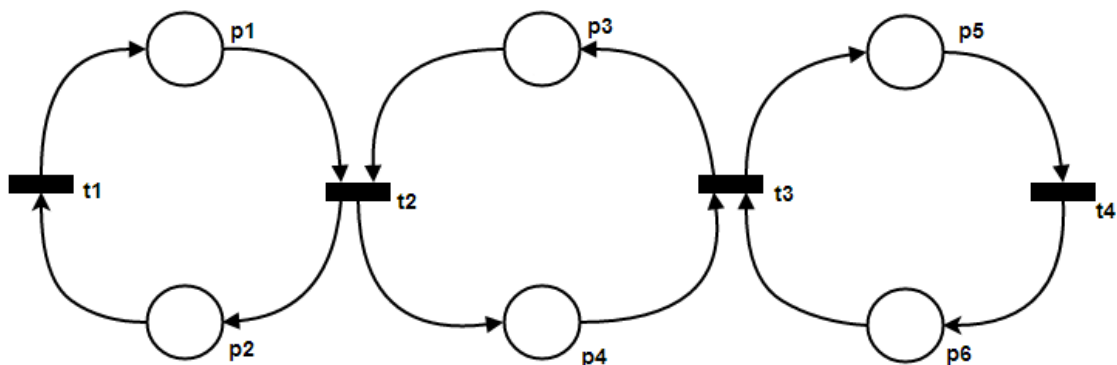


Figura 2.6: Exemplo do gráfico da RdP.

Esta RdP representa uma forma abstrata de modelar um sistema. A cada transição, estão associados lugares de entrada (lugares a partir dos quais saem os arcos dirigidos em direcção às transições) e lugares de saída (quando arcos dirigidos saem de transições para esses lugares). Por exemplo, na RdP da **Figura 2.6**, p5 e p6 são lugares de entrada da transição t4 e p2 e p4 são lugares de saída da transição t2.

As RdP também podem modelar a dinâmica de um sistema. Essa dinâmica é conseguida através do disparo das transições, onde existe produção/consumo de marcas, representadas por pontos pretos, presentes no interior dos lugares. Assim, a marcação de uma RdP é a distribuição das marcas pelos lugares, correspondente ao estado da RdP [6].

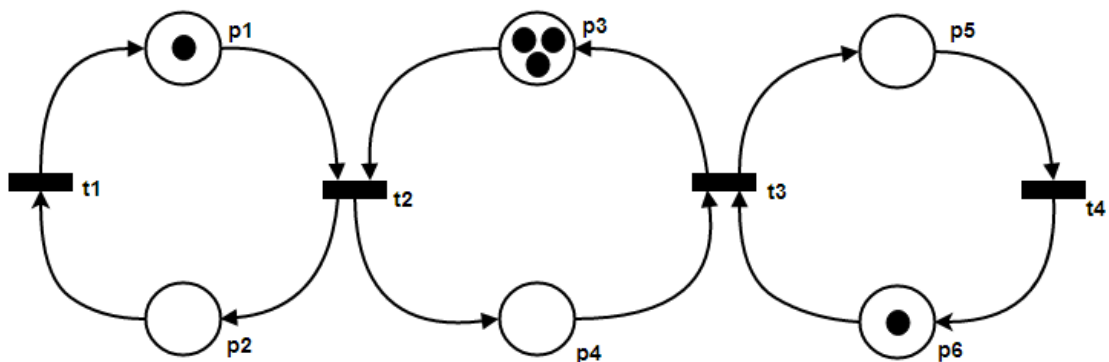


Figura 2.7: Exemplo de marcação da RdP da **Figura 2.6**.

Uma transição está habilitada a disparar quando existe pelo menos uma marca em todos os lugares de entrada dessa transição. Na **Figura 1.7**, verifica-se que apenas a transição *t2* está habilitada a disparar. A regra de disparo das RdP dita que uma das marcas presente em cada lugar de entrada dessa transição seja consumida/destruída e que seja produzida/criada uma marca em cada um dos lugares de saída dessa transição.

Assim, na **Figura 2.8**, está representada a evolução da RdP da **Figura 2.7**, após o disparo da transição *t2*:

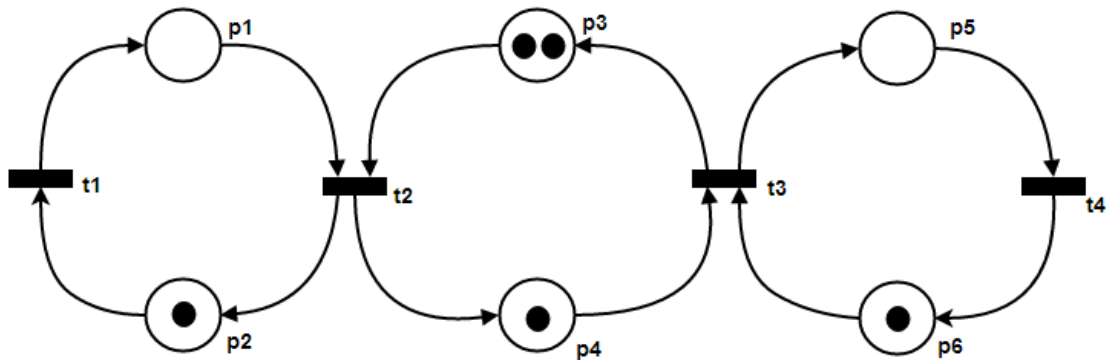


Figura 2.8: Evolução da marcação da RdP da **Figura 2.7**.

As transições e os lugares de entrada e de saída poderão ter diversos significados/interpretações. Assim, os lugares de entrada poderão ser interpretados como pré-condições, dados ou sinais de entrada, recursos que são precisos, buffers, condições, etc. As transições poderão significar eventos, passos de execução, tarefas, etc. Os lugares de saída poderão traduzir-se em pós-condições, dados ou sinais de saída, recursos livres, buffers, conclusões, etc [5].

Com o passar dos anos, as RdP providenciaram as bases para um largo espectro de linguagens visuais, normalmente chamadas de classes de Redes de Petri. Entre elas, identificam-se dois grandes grupos: autónomas e não-autónomas.

Uma RdP autónoma é a componente abstracta do modelo RdP do sistema, ou seja, estrutura de grafo e marcação. Uma RdP não-autónoma integra adicionalmente características que a ligam ao mundo exterior, como temporizações e sinais de entrada e de saída [6]. As RdP IOPT são um exemplo de RdP não-autónomas.

2.1.2.2 Modelação com Redes de Petri

Como referido anteriormente, as RdP são uma representação abstracta de um sistema. Na tentativa de se modelar um determinado sistema, torna-se necessário atribuir uma interpretação à RdP. Pretende-se assim, que essa interpretação seja uma representação fiel da realidade. Tendo como exemplo a RdP da **Figura 2.7**, apresenta-

se na **Figura 2.9**, uma possível interpretação da RdP modelando um sistema produtor-consumidor [6], tendo o armazém capacidade igual a três.

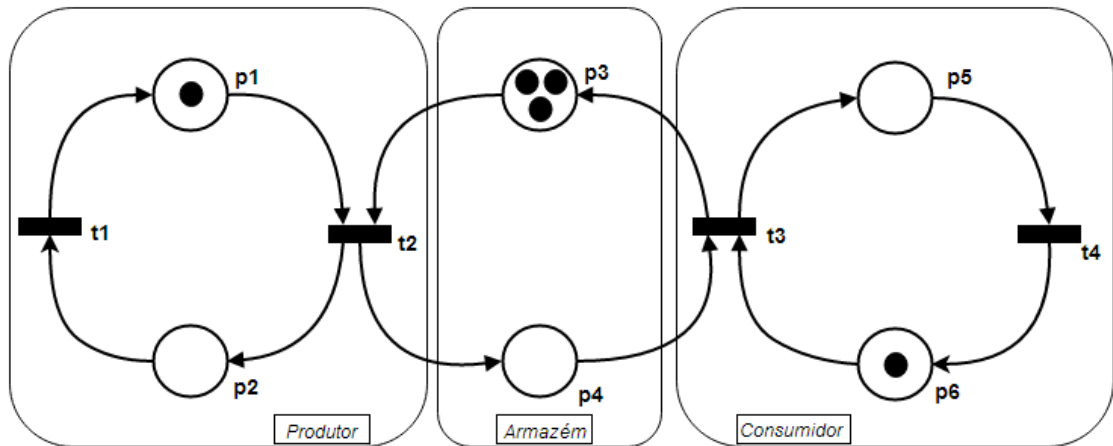


Figura 2.9: Rede de Petri de sistema produtor-consumidor com armazém com capacidade igual a três.

O ciclo de produção agrupa os lugares p1 e p2 e as transições t1 e t2. O ciclo de consumo é modelado pelos lugares p5 e p6 e pelas transições t3 e t4. O armazém é modelado pelos lugares p3 e p4, que representam, respectivamente, a capacidade do armazém e os produtos armazenados.

2.1.2.3 Classe IOPT

A classe de Redes de Petri Input Output Place Transition (IOPT), proposta em [7], são uma extensão às classes de RdP Lugar-Transição [8] mas com características não-autônomas. Esta classe foi implementada especificamente para dar suporte à implementação de controladores de sistemas embutidos. Sendo especialmente útil para modelação de controladores conduzidos por eventos discretos, onde sinais e eventos externos (de entrada e de saída) são muito comuns.

Os eventos e sinais permitem a especificação das interações entre um modelo do controlador (a rede) e o mundo exterior.

Os sinais de entrada são utilizados para obter informação do mundo exterior, por exemplo, para ler sinais de sensores, dos botões da interface com o utilizador ou sinais provenientes de outros sistemas.

Os sinais de saída podem ser utilizados, por exemplo, para manipular actuadores mecânicos, iluminar *Light Emitting Diodes* (LEDs) ou enviar informação para outros sistemas.

Um sinal digital é representado por um valor booleano e um sinal analógico é representado por um valor *range*¹ (intervalo).

Os eventos são normalmente associados a alterações no valor dos sinais.

Os eventos de entrada são desencadeados por alterações em sinais de entrada e os eventos de saída irão causar alterações em sinais de saída.

Um controlador IOPT pode esperar por um determinado evento de entrada e reagir de acordo, alterando o valor dos sinais de saída. Sistemas complexos podem ser implementados a partir de diversos sub-sistemas IOPT, comunicando entre si através de sinais e eventos.

Dá-se o nome de eventos autónomos a um sub-tipo de eventos de entrada, não associados a qualquer sinal de entrada, e que são utilizados exclusivamente para a comunicação entre sub-sistemas IOPT.

¹ “*range*”: valor inteiro que está compreendido num intervalo entre um valor inteiro mínimo e um valor inteiro máximo, inclusive.

Esta classe apresenta ainda as seguintes características:

- Eventos e sinais de entrada podem ser associados às transições.
- Eventos e sinais de saída podem ser associados às transições e lugares.
- Sinais de entrada e de saída podem ter valores booleanos ou valores de *range* (intervalo).
- Um lugar pode ter associado expressões (que representam condições), permitindo a atribuição de valores aos sinais de saída (em função de uma marcação específica), quando essas expressões forem verdadeiras.
- Os lugares têm um atributo *bound*, que corresponde ao limite de marcas.
- Uma transição dispara caso esteja habilitada e caso as condições externas associadas (eventos e sinais de entrada) sejam avaliados como verdadeiras.
- A evolução de uma rede IOPT também só é possível em instantes específicos no tempo, chamados *tics*, definidos por um relógio global externo. Ao período entre dois *tics*, dá-se o nome de passo de execução.
- Após o disparo, as transições podem originar eventos de saída, que por sua vez irão alterar os sinais a que estão associados.
- Para cada transição, é possível associar uma prioridade, que pode ser definida através de árbitros específicos, com o objectivo de resolução de conflitos. É também possível associar guardas que usam sinais externos (são condição necessária mas não suficiente para que a transição esteja habilitada) e eventos de entrada e de saída.
- O uso de uma linguagem de inscrição, com uma determinada sintaxe, permitindo especificar acções de saída associadas aos lugares, bem como, expressões algébricas, variáveis e funções para a especificação de guardas e condições de transição.

- Atribuição de pesos aos arcos.

2.2 IOPT-Tools

A IOPT-Tools é um ambiente de desenvolvimento integrado, que oferece um conjunto de ferramentas específico para as RdP da classe IOPT, com uma interface Web, e que dá suporte ao desenvolvimento e teste de controladores digitais.

As ferramentas incluem um editor gráfico de RdP, um módulo de verificação do modelo da RdP constituído por um gerador de espaço de estados, um visualizador do espaço de estados e um sistema de *queries*, bem como por um conjunto de ferramentas de geração automática de código (C, VHDL). O intercâmbio de informação do modelo da RdP IOPT é assegurado através do ficheiro PNML.

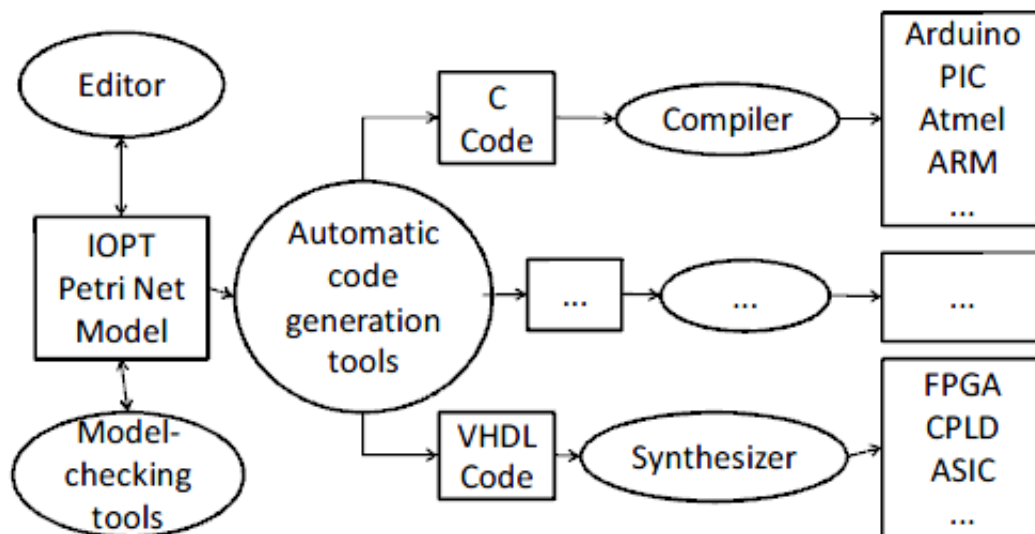


Figura 2.10: Adaptado de L. Gomes, F. Moutinho, F. Pereira 2013 [9].

Todas as ferramentas são executadas directamente no web browser do utilizador, a partir de princípios da tecnologia *Asynchronous JavaScript and XML* (AJAX) mas com a particularidade de que o armazenamento de ficheiros e o processamento de operações serem processados numa nuvem (cloud). Deste modo, minimiza o tráfego de dados da rede e permite o uso de equipamento terminal leve (*smartphone* ou *tablet*) no acesso às ferramentas. As IOPT-Tools foram testadas nas

versões mais recentes dos principais *browsers*, entre os quais, *Chrome*, *Firefox*, *Opera* e *Safari*, utilizando quer computadores pessoais quer tablets. Estão disponíveis online de forma gratuita [16].

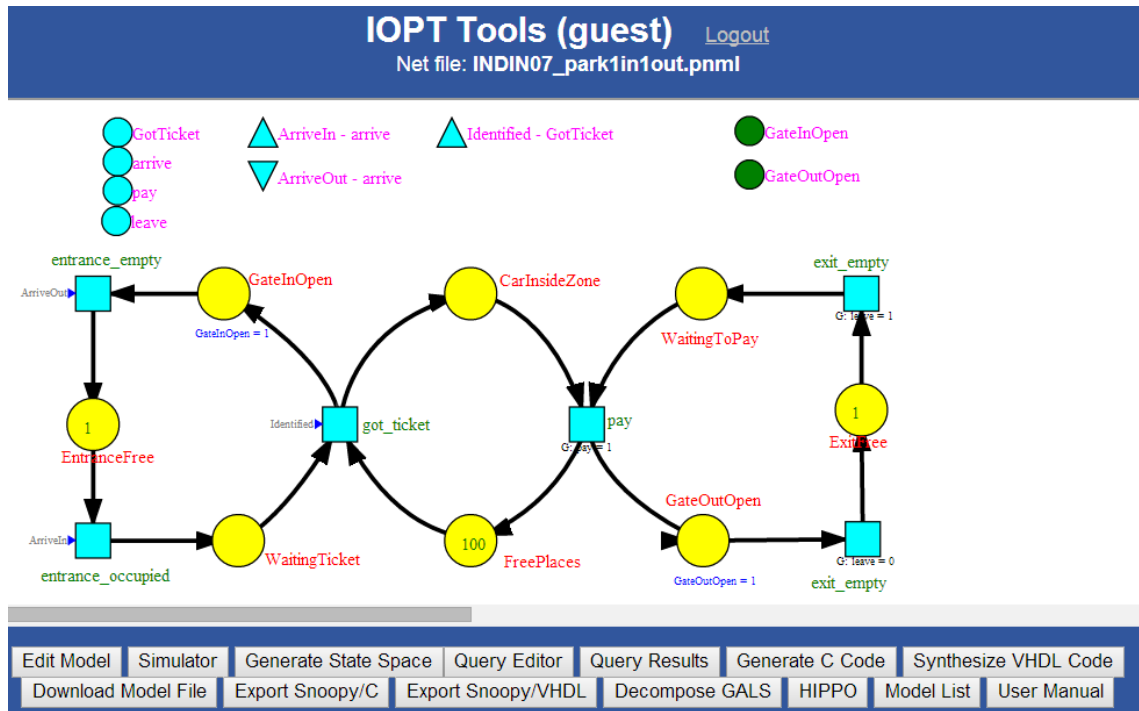


Figura 2.11: Aspecto visual da IOPT-Tools com as suas funcionalidades [16].

2.3 PNML

A *Petri Net Markup Language* (PNML) é um formato de representação de RdP baseado em *Extensive Markup Language* (XML).

Com o objectivo de se conseguir dar suporte a todos os tipos de RdP, a PNML foi uma das propostas que surgiu durante a Conferência Internacional sobre Aplicação e Teoria de Redes de Petri 2000 [10].

Por ser baseada em XML, a PNML é legível por pessoas e perfeitamente alterável utilizando qualquer editor de texto.

Todas as características comuns a todas as classes de redes de Petri ficam num ficheiro com extensão .pnml, e todas as características particulares das classes num ficheiro com extensão .pntd (*Petri Net Type Definition*). Assim, este formato padrão das RdP, conseguirá representar qualquer versão das RdP com qualquer tipo de extensão/classe, garantindo-se igualmente o máximo de informação possível da RdP.

A PNML foi utilizada nos diversos trabalhos associados às IOPT-Tools, como meio de ligação entre elas, garantindo-se assim o intercâmbio de informação. Foi igualmente utilizada no âmbito desta dissertação numa fase embrionária do Wave4IOPT, com o objectivo de se extrair informação do modelo de RdP IOPT (leitura de ficheiros .pnml).

2.4 JSON

JSON (JavaScript Object Notation - Notação de Objetos JavaScript) é uma formatação leve de troca de dados e está baseado num subconjunto de regras da linguagem de programação JavaScript [11], usadas na representação de objectos e *arrays* literais¹ (ver Exemplo 1).

Exemplo 1 - Exemplos de alguns tipos de literais em JavaScript:

"Ruben Lima" – string literal;

[1, 2, 3] – array literal;

{x:3, y:2} – objecto literal;

true – booleano literal.

¹ "literal", é um valor em JavaScript e contém informação de modo explícito e directo, ou seja, não se refer a informação de forma indirecta.

Para além de ter uma sintaxe muito semelhante ao JavaScript, o JSON é fácil de ler e escrever (para os seres humanos) e de interpretar e gerar (para as máquinas). Pode ser gerado a partir de qualquer linguagem de programação existente e é fácil de converter (*parse*), sendo, por isso, um formato ideal de transmissão de dados entre sistemas.

JSON é constituído por duas estruturas de dados universais [17]:

- Uma colecção de pares chave/valor. Em várias linguagens, isto é caracterizado como um objecto, *record*, estrutura (*struct*), dicionário, *hash table*, *keyed list*, ou *arrays* associativos.
- Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como um *array*, vector, lista ou sequência.

No formato JSON, os dados são apresentados da seguinte forma:

Valores – podem ser objectos, *arrays*, *strings* (delimitadas por aspas (“Rede”) ou plicas (“Petri”)), números, booleanos (*true* ou *false*), e o valor *null*.

Objeto – é um conjunto desordenado de pares chave/valor. Um objeto começa com “{” e termina com “}”. Cada nome é seguido por “:” e os pares chave/valor são seguidos por “,” (ver Exemplo 2).

Exemplo 2 – Exemplo de um objecto em JSON:

```
{  
  "nome": "SinalX",  
  "valorMin": 0,  
  "valorMax": 1,
```

```
    "comportamento": [0, 1, 0, 1, 1, 1, 0, 1, 0],  
    "estado": true  
}
```

Array – é uma colecção de valores ordenados. O array começa com "[" e termina com "]". Os valores são separados por ",".

Exemplo 3 – Exemplos de alguns tipos de *arrays* em JSON :

1. [1, 0, 3, 5]
2. ["SinalA", "SinalB", "SinalC"]
3. [[1, 0, 1], [2, 3, 1], [0, 0, 1]]

Para efeitos desta Dissertação, surgiu a necessidade de converter dados em JavaScript desde/para dados em JSON, sendo conseguida, respectivamente, através dos métodos *stringify* e *parse*.

Exemplo de conversão de objecto "*allWaveforms*" em JavaScript para *string* JSON (*serialization*):

```
var json_text = JSON.stringify( allWaveforms, null, 2 );
```

Exemplo de conversão de *string* JSON para um objecto "*allWaveforms*" JavaScript:

```
allWaveforms = JSON.parse( evt.target.result );
```

2.5 SVG

SVG é a abreviatura para *Scalable Vector Graphics*, uma linguagem baseada em XML, capaz de descrever de forma vectorial, desenhos e gráficos bi-dimensionais (num sistema de coordenadas X,Y) quer de forma estática, quer dinâmica ou animada.

Por ser uma linguagem baseada em vectores e não em píxeis, significa que são simples representações matemáticas de objectos gráficos. Esse facto, faz com que SVG seja infinitamente escalável, ou seja, uma imagem em SVG não perderá qualidade quando for ampliada. O mesmo não acontece com outros formatos de dados para imagens, como JPEG, PNG ou GIF.

SVG permite três tipos de objectos gráficos: formas geométricas simples (como linhas, rectângulos, círculos, elipses, polígonos) ou mais complexas (através de *paths* criados a partir de linhas, arcos, curvas, etc.), imagens e texto. Podem ser agrupados, alterados no seu aspecto visual e transformados (é possível mover, escalar, ou rodar os objectos gráficos sobre o sistema de coordenadas X,Y). Também é possível aplicar efeitos mais complexos, como criar padrões e aplicar máscaras em formas geométricas ou criar filtros em imagens.

Para informação mais detalhada sobre SVG consultar [13] e [18].

3

A Ferramenta desenvolvida: Wave4IOPT

Neste capítulo é apresentada a ferramenta Wave4IOPT implementada, que irá incorporar o conjunto da IOPT-Tools. São apresentadas as suas funcionalidades através de formalismos de modelação.

3.1. Descrição da ferramenta

A ferramenta Wave4IOPT representa o Editor e Visualizador Web de Formas de Onda específico para a classe IOPT das Redes de Petri.

A **Figura 3.1** representa a arquitetura geral do sistema:

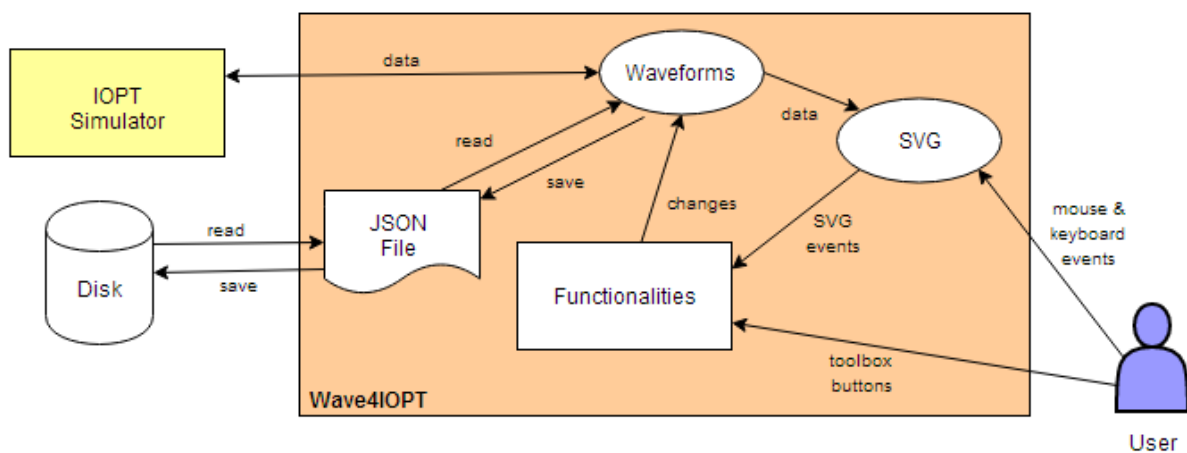


Figura 3.1: Arquitectura geral do sistema.

Na **Figura 3.1** estão identificadas as seguintes entidades:

Wave4IOPT – Editor e Visualizador de Formas de Onda Web para RdP IOPT. Representado por um rectângulo rosa.

User – Utilizador (entidade humana) que actua no Wave4IOPT, efectuando alterações ao nível da edição e/ou visualização das formas de onda. Representado por um boneco azul.

IOPT Simulator – Simulador das IOPT-Tools, representado por um rectângulo a amarelo, responsável pela simulação de RdP IOPT e do envio da informação dessa simulação directamente para um objecto JavaScript *allWaveforms* (entidade Waveforms) do Wave4IOPT, utilizando uma função para o efeito. Também poderá receber informação do Wave4IOPT, mais propriamente, proveniente da estrutura *allWaveforms*, após a conclusão das edições nas Formas de Onda efectuadas pelo *User*.

Disk – representa um disco interno ou externo ao computador pessoal do *User*.

JSON File – ficheiro JSON que poderá ser criado pelo Wave4IOPT e guardado em *Disk* ou ser lido a partir de *Disk* para o Wave4IOPT.

Waveforms – representa a estrutura que contém toda a informação de um conjunto de Formas de Onda (*allWaveforms*).

SVG – representa o documento SVG apresentado no ecrã visualizado pela entidade *User*. O *SVG* é sensível a eventos do rato e teclado provocados pelo *User* e associados a funcionalidades do Wave4IOPT.

Functionalities – representa todo o conjunto de funcionalidades inerentes à edição e visualização das Formas de Onda. Entidade responsável por fazer alterações na estrutura das Formas de Onda (*Waveforms*), bem como, de receber eventos do rato e teclado provenientes da entidade *SVG* e dos botões da *toolbox*.

O Wave4IOPT é *browser-based*. Significa que se pode acede-lo através da internet por um navegador (*browser*). Esta forma de implementação tem muitas vantagens quando comparada com a forma tradicional de implementar editores gráficos *desktop-based*. Nomeadamente, leva à redução de código na fase de implementação, à redução no tráfego de dados da rede quando se acede à ferramenta a partir de um computador pessoal, bem como, a possibilidade de se poder aceder utilizando um equipamento mais leve como um *smarthphone* ou *tablet*.

O Wave4IOPT foi implementado utilizando tecnologias Web como HTML, JavaScript, CSS, SVG e JSON.

3.1.1 Dados recebidos e enviados

O Wave4IOPT poderá receber dados provenientes de uma simulação de uma Rdp IOPT do Simulador, por duas vias, directa ou indirecta:

- Via directa, a partir de uma função do Simulador que fará com que seja lida toda a informação para um objecto JavaScript *allWaveforms* em memória.
- Via indirecta, caso o *User* abra um ficheiro JSON, localizado num disco interno ou externo ao computador pessoal.

Os dados recebidos contêm informação sobre algumas entidades da classe IOPT, nomeadamente: lugares (*places*), transições (*transitions*), eventos de entrada e de saída (*I/O events*) e sinais de entrada e de saída (*I/O signals*).

Essa informação, para ser devidamente aceite, terá que obedecer à estrutura predefinida *allWaveforms*, em que cada Forma de Onda é representada por nove campos. Por ordem: *name*, *type*, *subtype*, *minValue*, *maxValue*, *behaviourVector*,

viewMode, *numericalBase* e *visibility*. No **Anexo I** apresentam-se exemplos reais de um ficheiro JSON.

A título de exemplo, a figura seguinte pretende representar a estrutura de uma Forma de Onda “*EntranceFree(p_2)*”:

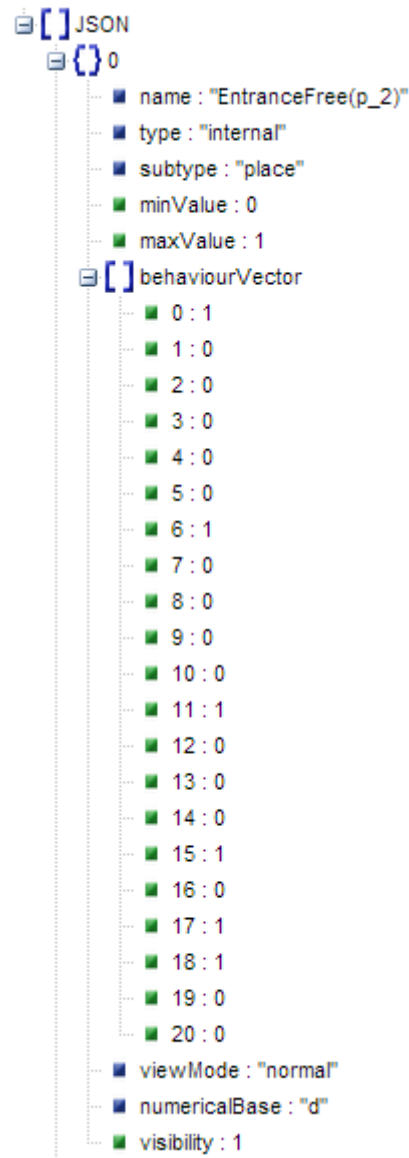


Figura 3.2: Estrutura de uma Forma de Onda¹

Em que:

name – representa o nome da Forma de Onda.

¹ representada através de um Visualizador de JSON disponível *online* [19]

type – representa o tipo de Forma de Onda. Poderá tomar um de três valores: “*internal*” (interno), caso a Forma de Onda seja do sub-tipo lugar ou transição; “*input*” (entrada), caso a Forma de Onda seja do sub-tipo sinal ou evento; “*output*” (saída), caso a Forma de Onda seja do sub-tipo sinal ou evento).

subtype – representa o sub-tipo da Forma de Onda. Poderá tomar um de quatro valores: “*place*”; “*transition*”; “*signal*”; “*event*”. Ou seja, respectivamente: lugar; transição; sinal; evento.

minValue – representa o valor mínimo que a Forma de Onda poderá tomar. Valor inteiro menor do que *maxValue*.

maxValue – representa o valor máximo que a Forma de Onda poderá tomar. Valor inteiro maior do que *minValue*.

behaviourVector – representa um *array* com os valores dessa Forma de Onda ao longo do tempo. O valor da posição *n* do *array* irá corresponder o valor no step *n+1* da Forma de Onda.

viewMode – representa o modo de visualização da Forma de Onda e poderá tomar um de quatro valores: “*normal*” para Formas de Onda que só podem ter valores booleanos, *false* (0) ou *true* (1); “*simpleComp*”, “*levelComp*” ou “*bit2bit*”, respectivamente para Formas de Onda no modo Compacto Simples, Compacto por Níveis ou Bit a Bit, que podem ter valores de intervalo (*range*).

numericalBase – representa a base numérica com que irá ser mostrado o valor da Forma de Onda num determinado passo (*step*). Poderá tomar um de quatro valores: “*d*” de decimal, “*b*” de binário, “*h*” de hexadecimal ou “*o*” de octal.

visibility – representa o estado da Forma de Onda no Wave4IOPT, ou seja, se é ou não visível no Wave4IOPT. Poderá tomar um de dois valores: 1 (visível) ou 0 (não visível).

Nota 1: Neste trabalho, sempre que for referido o termo “Forma de Onda”, será uma referência genérica a uma forma de onda representada no Wave4IOPT que, em particular, poderá ser uma destes quatro sub-tipos de Forma de Onda: um lugar, uma transição, um sinal ou um evento.

Nota 2: Os lugares e os sinais poderão ser representados graficamente no Wave4IOPT com valores inteiros num determinado intervalo (*range*). Os eventos e transições só poderão ser representados graficamente no Wave4IOPT com valores 0 ou 1 (representando respectivamente os valores booleanos *false* e *true*).

3.2. Funcionalidades

Neste sub-capítulo são descritas as funcionalidades implementadas neste trabalho. Essa descrição será apresentada, numa primeira fase, de uma forma sucinta, acompanhada por algumas imagens dos menus e/ou das próprias funcionalidades. Numa segunda fase, serão apresentadas as suas funcionalidades através de formalismos de modelação, como o diagrama de casos de uso, bem como, a descrição e modelo comportamental para cada caso de uso.

Este é o aspecto visual geral da ferramenta Wave4IOPT:

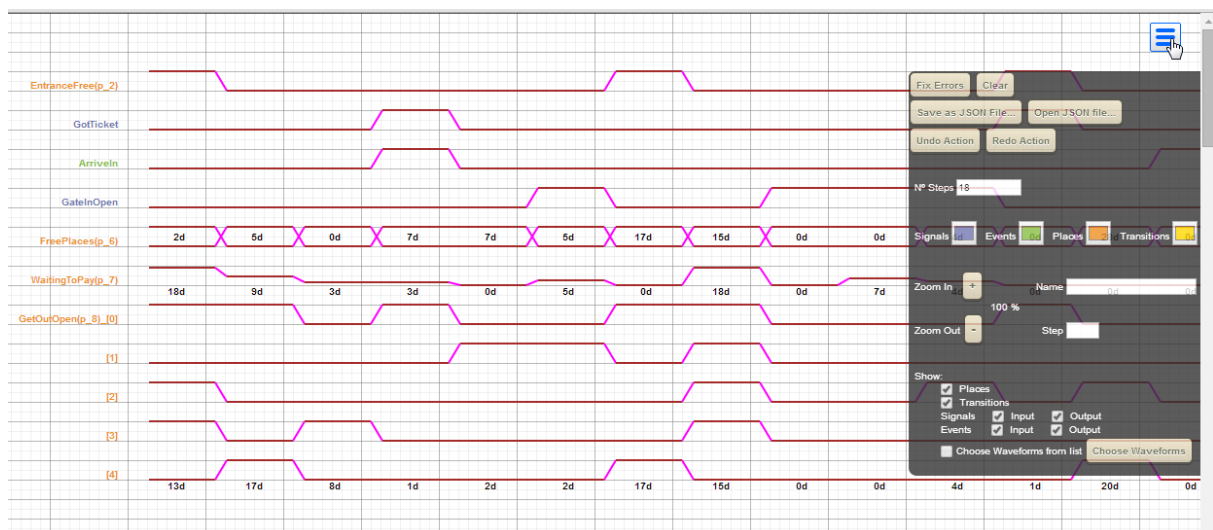


Figura 3.3: Aspecto visual do Wave4IOPT no browser Google Chrome.

Do lado esquerdo do ecrã estão listados os nomes das Formas de Onda. Por omissão, cada nome da Forma de Onda terá a seguinte cor, correspondente ao seu sub-tipo: laranja para os lugares (*places*), amarelo para as transições (*transitions*), verde para os eventos de entrada e saída (*I/O events*), azul para os sinais de entrada e saída (*I/O signals*) e preto para qualquer outro sub-tipo de Forma de Onda que não mencionado anteriormente (isto é, que não faça parte de uma RdP IOPT).

Imediatamente à direita de cada nome da Forma de Onda, está representado o comportamento temporal da Forma de Onda, para cada passo de execução.

Como já foi referido, as Formas de Onda podem ser representadas graficamente em quatro modos de visualização diferentes mas têm características comuns. Quando, ao longo do tempo, não existe variação do valor na Forma de Onda, as suas linhas horizontais são representadas:

- com cor castanha caso não exista qualquer erro no valor da Forma de Onda nesse passo (*step*).
- com cor vermelha caso exista um erro no valor da Forma de Onda nesse passo (*step*).

Quando, ao longo do tempo, existe uma variação do valor na Forma de Onda, as linhas oblíquas são representadas a cor-de-rosa.

No canto superior direito do ecrã estará sempre representado o botão (*burger button*) de opções do painel lateral de opções do Wave4IOPT. Ao ser clicado, com o botão esquerdo do rato, surgirá do lado direito do ecrã, um painel de opções com algumas funcionalidades.

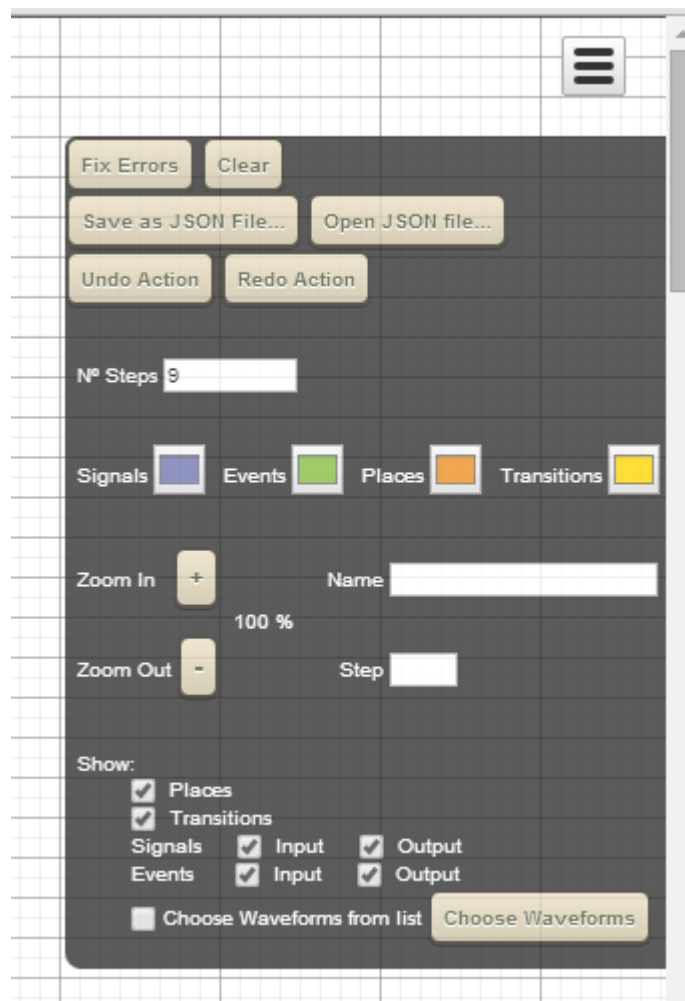


Figura 3.4: Aspecto visual do menu de opções lateral do Wave4IOPT no browser Google Chrome.

As funcionalidades dividem-se em funcionalidades de Edição da Forma de Onda e de Visualização no ecrã.

Na **Figura 3.4**, estão representados botões, caixas de selecção e caixas de texto associados a funcionalidades de edição ou de visualização.

O botão *Fix Errors*, está associado a um módulo simples de correcção de erros desta ferramenta, e tem a finalidade de corrigir todos os erros nos valores dos passos das Formas de Onda provenientes do ficheiro JSON. Considera-se a existência de um

erro no valor de um passo de uma Forma de Onda, quando esse passo da Forma de Onda apresenta como valor um número não inteiro e/ou fora do intervalo de valores permitidos para essa Forma de Onda.

O botão *Clear* tem a finalidade de remover todas as Formas de Onda que estejam visíveis no Wave4IOPT.

O botão *Save as JSON File...* permite a gravação dos dados de todas as Formas de Onda num ficheiro JSON, em qualquer disco interno ou externo ao computador pessoal.

O botão *Open JSON File...* permite a abertura de um ficheiro JSON com informação sobre o histórico de uma simulação de um conjunto de Formas de Onda ao longo do tempo.

Os botões *Undo Action* e *Redo Action* permitem, respectivamente, recuar e avançar uma acção efectuada pelo Utilizador no Wave4IOPT.

Nº Steps é representado por uma caixa de texto editável onde se especifica o número de passos (steps) que o Utilizador pretende visualizar.

Signals, Events, Places e *Transitions* representam botões de selecção de cor para os nomes dos sinais, eventos, lugares e transições, respectivamente.

Os botões *Zoom In* e *Zoom Out* correspondem às funcionalidades de ampliação e redução das formas de onda apresentadas na ferramenta. Poderão tomar valores entre os 25% e os 175%.

O *Name* e *Step* correspondem respectivamente ao nome da Forma de Onda e o número do passo. Esta funcionalidade de visualização só é possível de ser observada quando o Utilizador coloca o ponteiro do rato em cima dos segmentos de linhas castanhas horizontais.

A secção *Show* está representada por um conjunto de caixas de selecção, cada uma representativa de um tipo ou sub-tipo de Forma de Onda. A finalidade desta

funcionalidade é de permitir ao Utilizador observar somente os tipos e sub-tipos de Forma de Onda que pretende visualizar.

No caso do Utilizador pretender escolher um a um as Formas de Onda que pretende observar, terá que seleccionar a caixa *Choose Waveforms from list* de modo a habilitar a utilização do botão *Choose Waveforms*. Após pressionado este botão, aparecerá um painel no topo do Wave4IOPT, como é indicado na **Figura 3.5**:

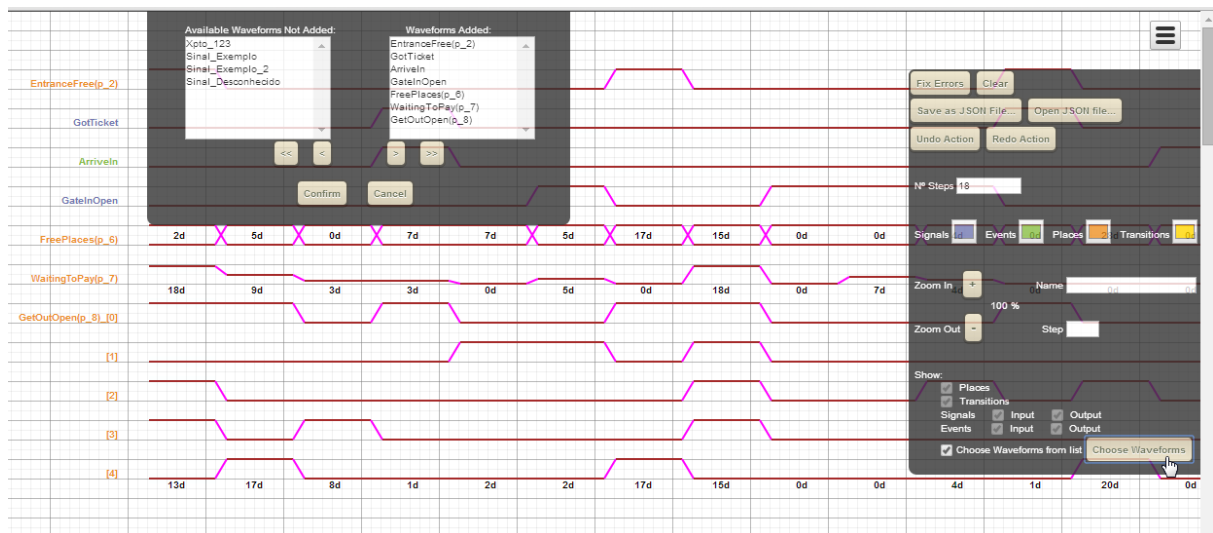


Figura 3.5: Aspecto visual do Wave4IOPT após pressionar do botão *Choose Waveforms*.

Observando mais em pormenor, este painel é constituído por duas listas de Formas de Onda: à esquerda as Formas de Onda que existem mas que não estão a ser visualizadas e à direita as Formas de Onda que estão a ser visualizadas no Wave4IOPT (**Figura 3.6**).

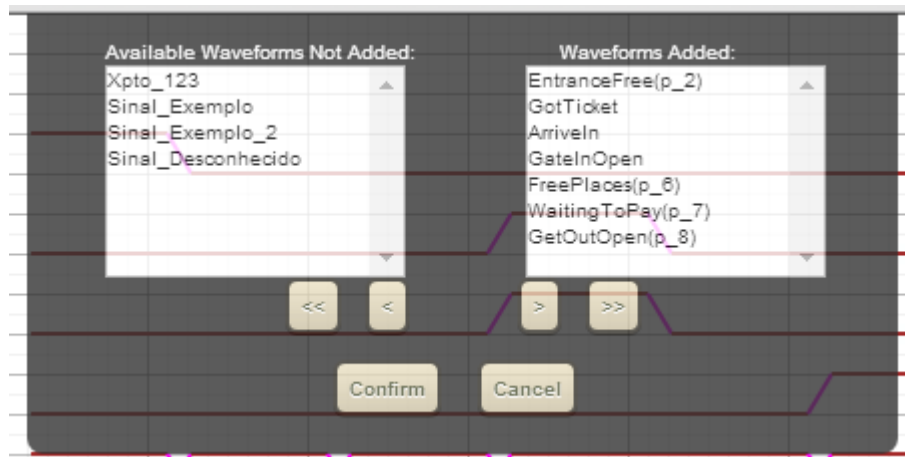


Figura 3.6: Aspecto visual do painel de topo do Wave4IOPT.

Cada Forma de Onda visível no Wave4IOPT tem um Menu Individual associado.

Quando o utilizador passa com o rato por cima do nome de uma Forma de Onda visível no Wave4IOPT, aparece o símbolo +. Com esse símbolo visível, ao carregar-se com o botão direito do rato, a Forma de Onda fica assinalada através de um rectângulo rosa e o Menu Individual da Forma de Onda aparece. Caso a Forma de Onda só possa ter valores booleanos ao longo do tempo, o Menu terá o seguinte aspecto:

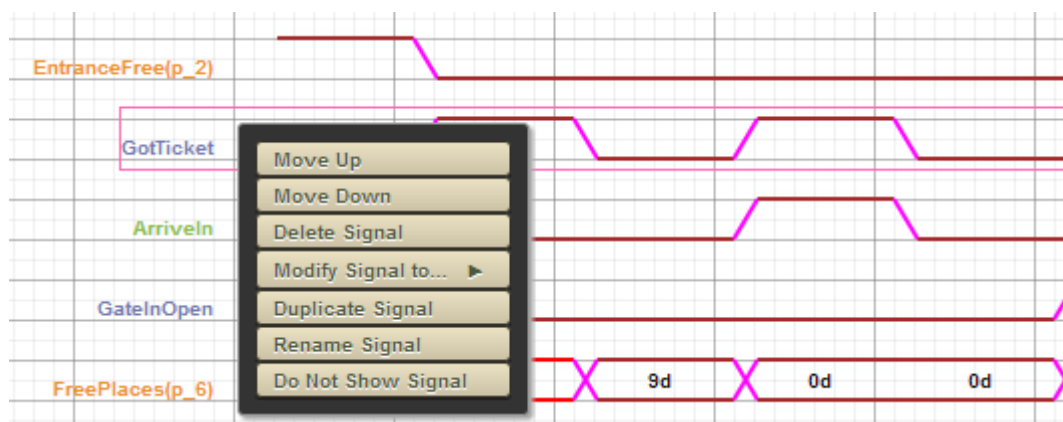


Figura 3.7: Menu individual da Forma de Onda.

Assim, como se pode observar na **Figura 3.7**, neste caso, o Menu está associado à Forma de Onda “*GotTicket*” e tem as seguintes opções:

Move Up – A Forma de Onda indicada será movida uma posição, no sentido ascendente, no conjunto de Formas de Onda visualizadas no Wave4IOPT.

Move Down – A Forma de Onda indicada será movida uma posição, no sentido descendente, no conjunto de Formas de Onda visualizadas na ferramenta.

Delete Waveform – A Forma de Onda indicada será removida do Wave4IOPT e deixará de existir na lista de Formas de Onda existentes, deixando assim de estar disponível para edição e visualização.

Duplicate Waveform – A Forma de Onda indicada será duplicada no Wave4IOPT aparecendo na última posição do ferramenta. O nome da Forma de Onda duplicada terá que ser válido e será dado pelo Utilizador, a pedido do Wave4IOPT, através de uma caixa de diálogo. Considera-se uma Forma de Onda com um nome válido se o Utilizador introduzir um nome diferente de uma *string* vazia, diferente de uma *string* com espaços e diferente de algum nome de Forma de Onda no Wave4IOPT. No caso de se introduzir um nome inválido, a ferramenta irá pedir (infinitamente) ao Utilizador o nome para a Forma de Onda duplicada até que este seja um nome válido, a não ser que o Utilizador aborte a operação. Para abortar a operação, basta o Utilizador carregar no botão “Cancel” ou de fecho da caixa de diálogo.

Rename Waveform – Será pedido ao Utilizador que dê um novo nome à Forma de Onda indicada. Se o novo nome for válido então a Forma de Onda ficará com o novo nome dado pelo Utilizador. Um nome válido é um nome diferente de uma *string* vazia, diferente de uma *string* com espaços e diferente de algum nome que já exista. Excepção será feita se o Utilizador mantiver o nome original da Forma de Onda ou se carregar no botão “Cancel”, ou seja, a Forma de Onda não terá o seu nome alterado.

Do Not Show Waveform – A Forma de Onda indicada deixará de estar visível no Wave4IOPT, apesar de ainda existir na lista de Formas de Onda existentes.

Modify Waveform to... – Esta opção só é visível para uma Forma de Onda do tipo *input*, ou seja, para eventos e sinais de entrada, e permite modificar a Forma de Onda. Para isso, o Utilizador terá que seleccionar uma das opções do sub-menu apresentado na **Figura 3.8**:

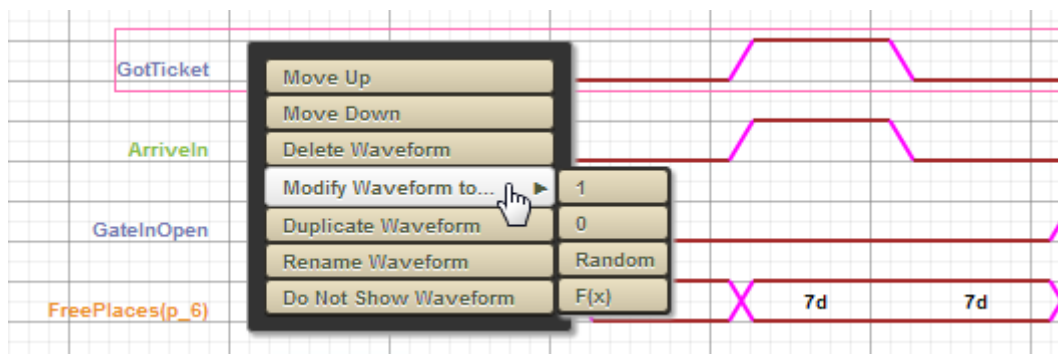


Figura 3.8: Sub-menu do Modificar Forma de Onda.

As opções deste sub-menu são as seguintes:

1 – A Forma de Onda indicada terá sempre o valor “1” ao longo do tempo.

0 – A Forma de Onda indicada terá sempre o valor “0” ao longo do tempo.

Random – A Forma de Onda indicada terá valores aleatórios ao longo do tempo. Esses valores aleatórios têm que estar compreendidos entre os valores mínimo e máximo (inclusive) permitidos para essa Forma de Onda.

F(x) – A Forma de Onda indicada terá como valores os resultados de uma fórmula escrita pelo Utilizador. Esses valores só serão válidos caso estejam compreendidos entre os valores mínimo e máximo (inclusive) permitidos para essa Forma de Onda.

Waveform view mode – Caso se trate de uma Forma de Onda com valores num intervalo (*range*), o Menu Individual da Forma de Onda terá adicionalmente esta

funcionalidade de Visualização. Permitirá que esta Forma de Onda possa ser apresentada de 3 modos diferentes, através do sub-menu apresentado na **Figura 3.9**:

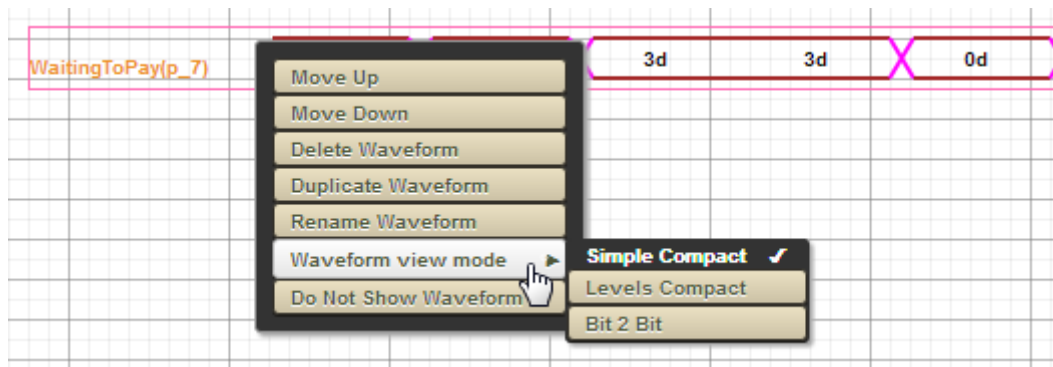


Figura 3.9: Sub-menu dos modos de visualização das Formas de Onda.

As opções deste sub-menu são as seguintes:

Simple Compact – A Forma de Onda indicada terá o aspecto da **Figura 3.10**, sendo representada no modo Compacto Simples.

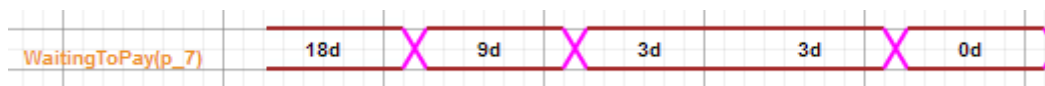


Figura 3.10: Forma de Onda no modo Compacto Simples.

Tal como visualizado na **Figura 3.10**, caso exista uma variação de valor de dois passos consecutivos, essa variação será representada por um X cor-de-rosa, entre os dois passos. Por exemplo, entre o passo 1 (com valor 18 em decimal) e o passo 2 (com valor 9 em decimal).

Caso contrário, sempre que dois ou mais passos consecutivos apresentem o mesmo valor, as linhas castanhas horizontais estarão unidas. Por exemplo, entre o passo 3 (com valor 3) e passo 4 (igualmente com valor 3).

Levels Compact – A Forma de Onda indicada é representada no modo Compacto por Níveis na **Figura 3.11**. Cada nível corresponde ao valor da Forma de Onda nesse passo (*step*).

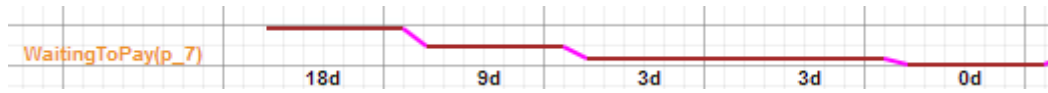


Figura 3.11: Forma de Onda no modo Compacto por Níveis.

A sua representação gráfica corresponde à percentagem que o valor ocupa, entre os valores mínimo e máximo permitidos para essa Forma de Onda, representados pelas linhas cinzentas horizontais de baixo e de cima, respectivamente. Assim, para esta Forma de Onda *WaitingToPay(p_7)*, os valores mínimo e máximo são, respectivamente, 0d (zero em decimal) e 20d (vinte em decimal).

Bit 2 Bit – A Forma de Onda no modo Bit a Bit contém uma ou mais formas de onda. Cada uma dessas as formas de onda corresponde a um bit dessa Forma de Onda no modo Bit a Bit e contém apenas valores booleanos (true ou false). Na **Figura 3.12** observa-se, de cima para baixo, as formas de onda do bit mais significativo até ao bit menos significativo.

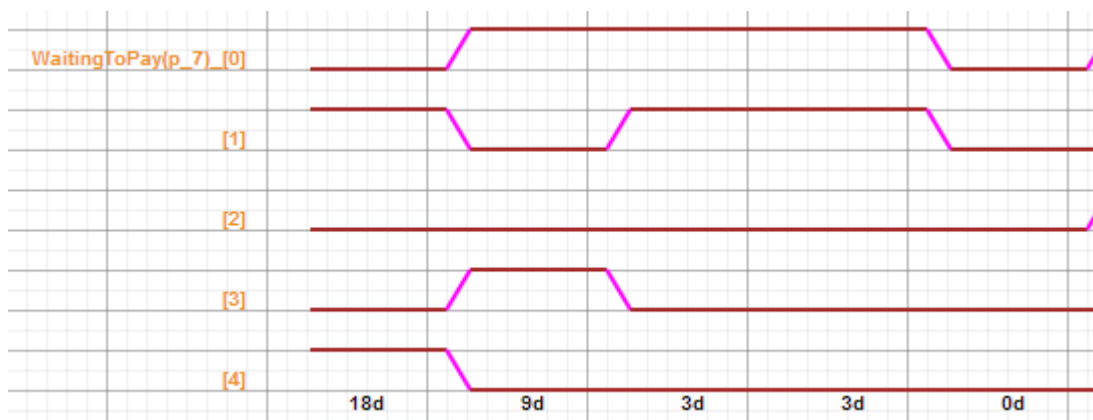


Figura 3.12: Forma de Onda no modo Bit a Bit.

Assim, no exemplo da **Figura 3.12**, o valor 18d (dezoito em decimal) no passo 1 da Forma de Onda *WaitingToPay(p_7)*, representa o somatório do produto dos valores das Formas de Onda nesse passo por 2^{Bit} . Ou seja, no passo 1, o valor 18 é calculado da seguinte forma: $0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 = 18$.

Estes três modos de visualização de Formas de Onda explicados anteriormente, que têm valores de intervalo (*range*), têm esses mesmos valores visíveis para cada passo de execução. Por omissão, esses valores são mostrados em base decimal. Contudo, através do Menu Bases Numéricas, é possível visualizar os valores em mais três bases: em binário, hexadecimal e octal. Para além disso, caso o tipo de Forma de Onda o permita, este menu apresentará uma opção adicional *Change Value*, que possibilita a alteração do valor nesse passo.

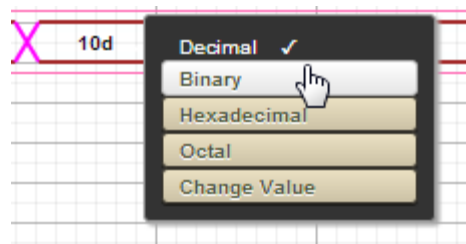


Figura 3.13: Menu Bases Numéricas.

Incrementar / Decrementar valores nos passos – O valor de um passo numa Forma de Onda pode ser incrementado / decrementado uma unidade através de uma acção de clique do rato. Essa acção poderá ser apenas efectuada em parte das linhas horizontais, que será indicada ao Utilizador quando fôr visível o ponteiro do rato da mão, em cima dessas linhas horizontais.



Figura 3.14: Antes do Incrementar de uma Forma de Onda de um evento.



Figura 3.15: Depois do Incrementar de uma Forma de Onda de um evento.

Numa Forma de Onda, o incremento / decremento do valor de um passo é obtido a partir do clique do botão esquerdo do rato. A exceção será o decremento de uma Forma de Onda no modo Compacto por Níveis, obtido através do clique do botão direito do rato. Esta exceção tem ainda a particularidade de só ser possível de ser executada caso não haja em memória uma cópia de uma Forma de Onda. Ou seja, a funcionalidade de copiar/colar tem prioridade em relação ao decrementar de uma Forma de Onda no modo Compacto por Níveis.

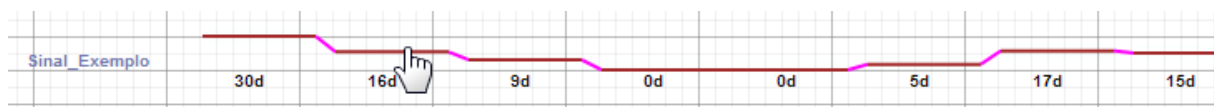


Figura 3.16: Antes do Decrementar de uma Forma de Onda no modo Compacto por Níveis.

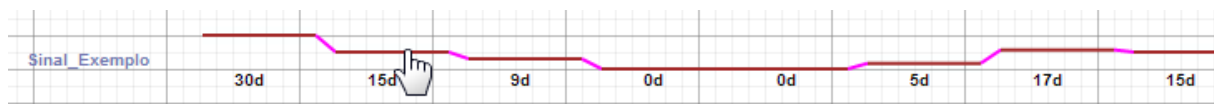


Figura 3.17: Depois do Decrementar de uma Forma de Onda no modo Compacto por Níveis.

Incrementar / Decrementar com CTRL e com CTRL + ALT – Também é possível incrementar/decrementar em mais do que uma unidade o valor num passo, bem como, nos passos seguintes. Assim, existem as funcionalidades especiais Incrementar com CTRL, Decrementar com CTRL, Incrementar com CTRL + ALT e Decrementar com CTRL + ALT.

Estas funcionalidades especiais obtêm-se quando o clique do botão do rato é acompanhado com o pressionar da tecla CTRL ou das teclas CTRL e ALT.

No caso do Incrementar/Decrementar com CTRL, o incremento/decremento não será de apenas uma unidade mas, respectivamente, para o valor máximo/mínimo permitidos para essa Forma de Onda. Estas acções serão válidas para o passo onde se executa o clique, bem como, para todos os passos seguintes até que dois passos consecutivos tenham, respectivamente, o valor mínimo permitido para essa Forma de Onda seguido de valor maior do que o mínimo permitido para essa Forma de Onda.

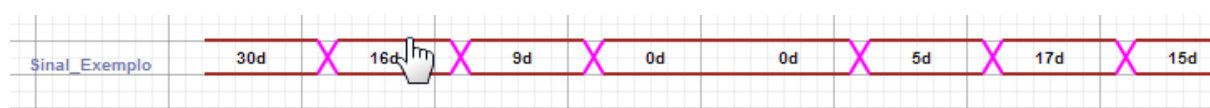


Figura 3.18: Antes do Incrementar com CTRL de uma Forma de Onda no modo Compacto Simples.

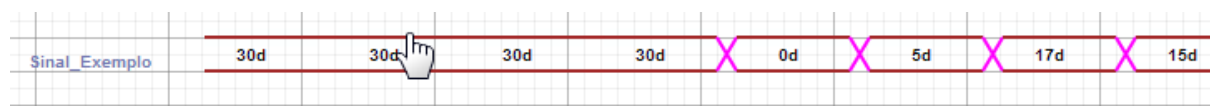


Figura 3.19: Depois do Incrementar com CTRL de uma Forma de Onda no modo Compacto Simples.

Nas **Figuras 3.18 e 3.19** está representado um *Sinal_Exemplo*, com valor mínimo de 0 e valor máximo de 30, antes e após o Incrementar com CTRL. Assim, na **Figura 3.19**, observa-se o resultado após se ter carregado na tecla CTRL do teclado e clicado com o botão esquerdo do rato em cima da linha horizontal superior do passo 2 do *Sinal_Exemplo*. Deste modo, desde o passo 2 até ao passo 4 (inclusive) a Forma de Onda apresenta o valor máximo permitido para essa Forma de Onda (neste caso 30 em decimal). Esta funcionalidade teve efeito até ao passo 4 pois os passos consecutivos 5 e 6 apresentam, respectivamente, o valor mínimo (valor 0) e valor superior ao mínimo (neste caso 5).

Caso o *Sinal_Exemplo*, por hipótese, não apresentasse dois passos consecutivos com valor mínimo e valor superior ao mínimo, a funcionalidade teria efeito desde o passo 2 até ao final dessa Forma de Onda. Assim, a partir do passo 2, todos os passos

seguintes teriam o valor máximo 30 (efeito igual à funcionalidade Incrementar com CTRL + ALT).

No caso das funcionalidades Incrementar/Decrementar com CTRL + ALT, o incremento/decremento será, respectivamente, para o valor máximo/mínimo permitidos para essa Forma de Onda. Esta acção será válida para o passo onde se executa o clique, bem como, para todos os passos seguintes dessa Forma de Onda.



Figura 3.20: Depois do Incrementar com CTRL + ALT de uma Forma de Onda de um evento.

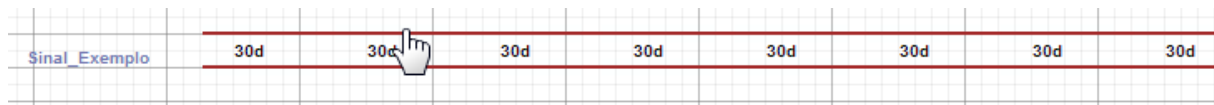


Figura 3.21: Depois do Incrementar com CTRL + ALT numa Forma de Onda de um sinal no modo Compacto Simples.

Nas **Figuras 3.20 e 3.21** estão, representados respectivamente uma Forma de Onda de um evento com valores booleanos (*ArriveIn*) e uma Forma de Onda de um sinal no modo Compacto Simples (*Sinal_Exemplo*) após o Incrementar com CTRL + ALT no passo 2. Assim, a partir do passo 2, todos os passos terão o valor máximo permitido para essas Formas de Onda (neste caso, 1 e 30, respectivamente).

Copiar e Colar os valores dos passos – Outra das funcionalidades de edição da ferramenta, é a de copiar o comportamento de uma determinada Forma de Onda durante N passos para, posteriormente, se colar esse comportamento no mesmo ou noutra Forma de Onda, a partir de determinado passo à escolha do Utilizador.

Nas **Figuras 3.22, 3.23, 3.24 e 3.25**, estão representadas as várias etapas das funcionalidades Copiar e Colar valores dos passos.

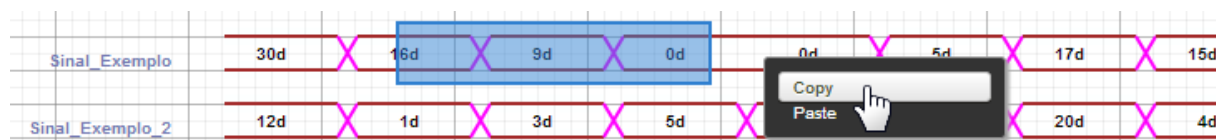


Figura 3.22: Seleção dos passos a serem copiados.

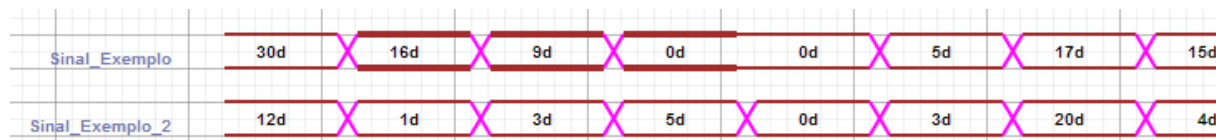


Figura 3.23: Aspecto visual após uma cópia dos valores.

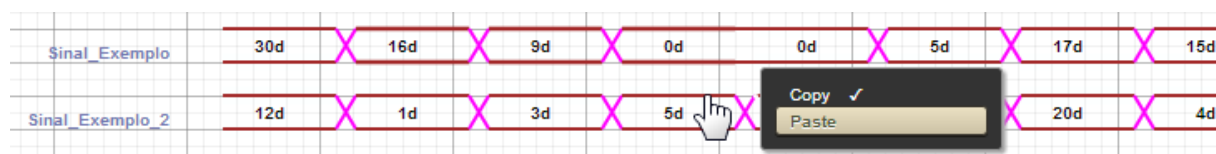


Figura 3.24: Antes da operação colar.

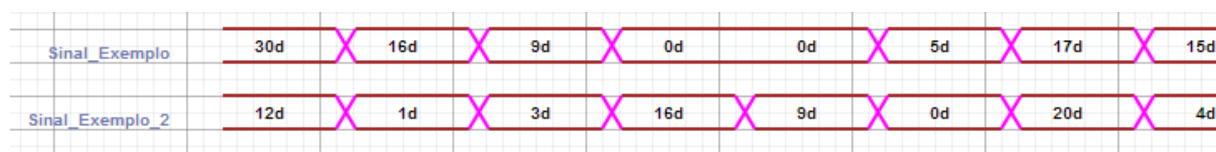


Figura 3.25: Após conclusão da operação colar.

Como observável nas **Figuras 3.22 e 3.23**, é efectuada a cópia da Forma de Onda do Sinal_Exemplo, desde o passo 2 ao passo 4 (inclusive), ficando bem visível quais os passos copiados após a cópia.

Posteriormente, o Utilizador carrega com o botão direito do rato em cima de uma das linhas horizontais do passo 4 do Sinal_Exemplo_2, de modo a poder colar o

comportamento da onda (**Figuras 3.24 e 3.25**). Note-se que só é possível colar o comportamento da onda uma vez. Isto significa que, após finalizada a colagem, a forma de onda anteriormente copiada deixará de estar em memória.

3.2.1. Diagrama de Casos de Uso

Na **Figura 3.26**, apresenta-se o Diagrama de Casos de Uso do Wave4IOPT. Na primeira metade do diagrama, estão representados os casos de uso relacionados com as funcionalidades de Edição do Wave4IOPT e na segunda metade as funcionalidades de Visualização:

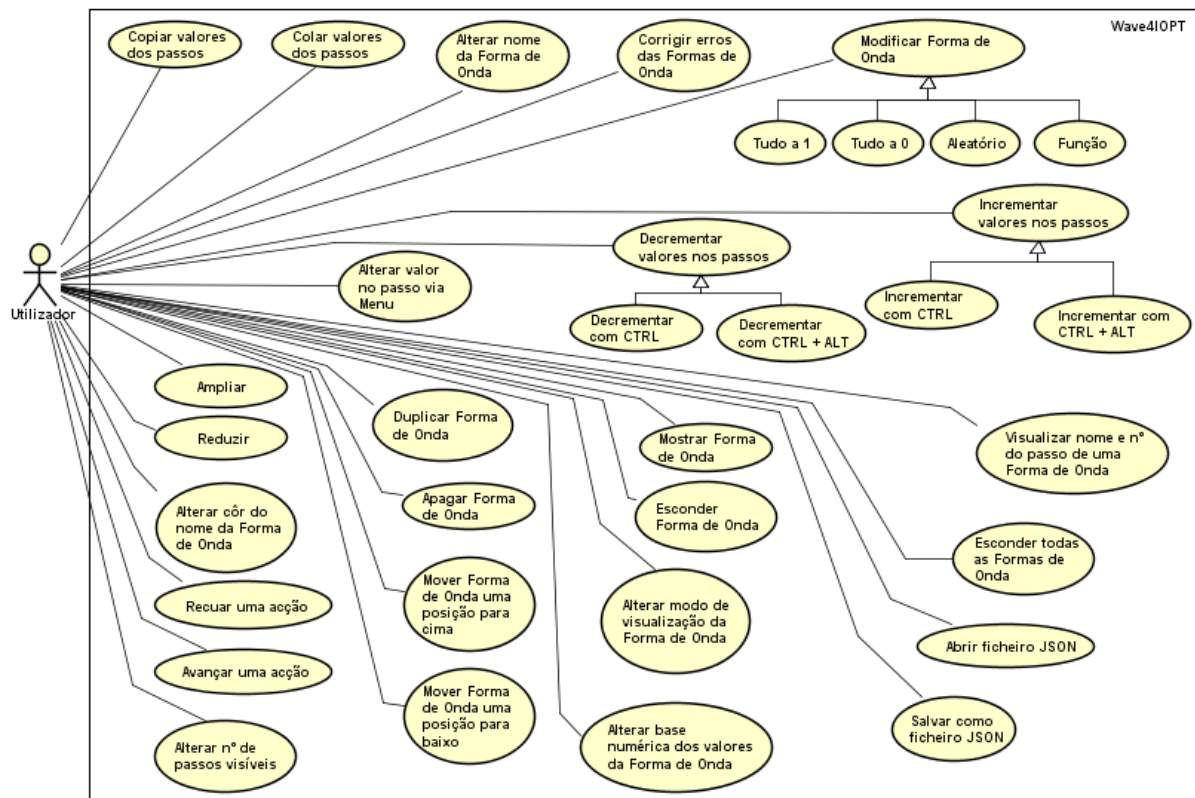


Figura 3.26: Diagrama de Casos de Uso do Wave4IOPT.

3.2.2. Descrição dos Casos de Uso

Com o propósito de descrever com maior detalhe todas as funcionalidades da ferramenta implementada, descrevem-se seguidamente os respectivos Casos de Uso:

Tabela 3.1 – Descrição de Caso de Uso Incrementar valores nos passos.

Título do Caso de Uso	Incrementar valores nos passos.
Pré-condições	<p>Tem que existir pelo menos uma Forma de Onda de um evento ou sinal de entrada visível no Wave4IOPT.</p> <p>Número de Passos (<i>Nº of Steps</i>) visíveis no Wave4IOPT tem que ser maior ou igual a 1.</p> <p>Valor actual no passo, tem que ser inferior ao valor máximo permitido para a Forma de Onda de um evento ou sinal de entrada onde se faz o clique.</p> <p>Caso exista um erro no valor actual no passo (assinalado a cor vermelha no Wave4IOPT), a anterior pré-condição não terá que se verificar.</p> <p>No caso de ser uma <i>Forma de Onda no modo Compacto Simples</i>, o clique com o botão esquerdo do rato terá que ser efectuado na linha horizontal de cima dessa Forma de Onda.</p>
Pós-condições	<p>A Forma de Onda de um evento ou sinal de entrada, nesse passo (<i>step</i>), passará a ter o valor incrementado em uma unidade, em relação ao que tinha anteriormente.</p> <p>No caso de ser um <i>Forma de Onda no modo Bit a Bit</i>, o valor do passo será incrementado em 2^{Bit}, sendo <i>Bit</i> o nº do Bit dessa Forma de Onda.</p>
Sumário	Utilizador faz um clique, com o botão esquerdo do rato, num determinado passo (<i>step</i>), em cima da linha de uma Forma de Onda de um evento ou sinal de entrada.

Actor	Utilizador.
Cenário principal	1. Clicar com o botão esquerdo do rato, num determinado passo (<i>step</i>), em cima da linha de uma Forma de Onda de um evento ou sinal de entrada, quando o icon do rato fôr uma mão.
Cenários alternativos	

Tabela 3.2 – Descrição de Caso de Uso Incrementar com CTRL.

Título do Caso de Uso	Incrementar com CTRL.
Pré-condições	<p>Tem que existir pelo menos uma Forma de Onda de um evento ou sinal de entrada visível no Wave4IOPT.</p> <p>Número de Passos (<i>Nº of Steps</i>) visíveis no Wave4IOPT tem que ser maior ou igual a 1.</p> <p>Valor actual no passo, tem que ser inferior ao valor máximo permitido para a Forma de Onda de um evento ou sinal de entrada onde se faz o clique.</p> <p>Caso exista um erro no valor actual no passo (assinalado a cor vermelha no Wave4IOPT), a anterior pré-condição não terá que se verificar.</p> <p>No caso de ser uma <i>Forma de Onda no modo Composto Simples</i>, o clique com o botão esquerdo do rato terá que ser efectuado na linha horizontal de cima dessa Forma de Onda.</p>
Pós-condições	<p>A Forma de Onda de um evento ou sinal de entrada, nesse passo (<i>step</i>), passará a ter o valor máximo permitido para essa Forma de Onda de um evento ou sinal de entrada.</p> <p>Os passos (<i>steps</i>) seguintes dessa Forma de Onda terão igualmente esse valor máximo, até que dois passos</p>

	consecutivos apresentem, respectivamente, valor mínimo permitido para essa Forma de Onda e valor maior que o mínimo permitido para essa Forma de Onda.
Sumário	Utilizador carrega na tecla CTRL do teclado e faz um clique, com o botão esquerdo do rato, num determinado passo (<i>step</i>), em cima da linha de uma Forma de Onda de um evento ou sinal de entrada.
Actor	Utilizador.
Cenário principal	1. Carregar na tecla CTRL do teclado e fazer um clique com o botão esquerdo do rato. O clique é efectuado num determinado passo (<i>step</i>), em cima da linha de uma Forma de Onda de um evento ou sinal de entrada, quando o icon do rato fôr uma mão.
Cenários alternativos	Operação não é permitida numa <i>Forma de Onda no modo Bit a Bit</i> : 1. a) Caso o clique tenha sido numa das Formas de Onda que constituem uma <i>Forma de Onda no modo Bit a Bit</i> , a operação não será permitida. Será apresentada uma mensagem no Wave4IOPT, informando o Utilizador do descrito anteriormente.

Tabela 3.3 – Descrição de Caso de Uso Incrementar com CTRL + ALT.

Título do Caso de Uso	Incrementar com CTRL + ALT.
Pré-condições	Tem que existir pelo menos uma Forma de Onda de um evento ou sinal de entrada visível no Wave4IOPT. Número de Passos (<i>Nº of Steps</i>) visíveis no Wave4IOPT tem que ser maior ou igual a 1.

	<p>Valor actual no passo, tem que ser inferior ao valor máximo permitido para a Forma de Onda de um evento ou sinal de entrada onde se faz o clique.</p> <p>Caso exista um erro no valor actual no passo (assinalado a cor vermelha no Wave4IOPT), a anterior pré-condição não terá que se verificar.</p> <p>No caso de ser uma <i>Forma de Onda no modo Composto Simples</i>, o clique com o botão esquerdo do rato terá que ser efectuado na linha horizontal de cima dessa Forma de Onda.</p>
Pós-condições	<p>A Forma de Onda de um evento ou sinal de entrada, nesse passo (<i>step</i>), passará a ter o valor máximo permitido para essa Forma de Onda.</p> <p>Todos os passos (<i>steps</i>) seguintes dessa Forma de Onda terão igualmente esse valor máximo.</p>
Sumário	Utilizador carrega nas teclas CTRL e ALT do teclado e faz um clique, com o botão esquerdo do rato, num determinado passo (<i>step</i>), em cima da linha de uma Forma de Onda de um evento ou sinal de entrada.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. Carregar nas teclas CTRL e ALT do teclado e fazer um clique com o botão esquerdo do rato. O clique é efectuado num determinado passo (<i>step</i>), em cima da linha de uma Forma de Onda de um evento ou sinal de entrada, quando o icon do rato fôr uma mão.
Cenários alternativos	<p>Operação não é permitida numa <i>Forma de Onda no modo Bit a Bit</i>:</p> <ol style="list-style-type: none"> 2. a) Caso o clique tenha sido numa das Formas de

	Onda que constituem uma <i>Forma de Onda no modo Bit a Bit</i> , a operação não será permitida. Será apresentada uma mensagem no Wave4IOPT, informando o Utilizador do descrito anteriormente.
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabela 3.4 - Descrição de Caso de Uso Decrementar valores nos passos.

Título do Caso de Uso	Decrementar valores nos passos.
Pré-condições	<p>Tem que existir pelo menos uma Forma de Onda de um evento ou sinal de entrada visível no Wave4IOPT.</p> <p>Número de Passos (<i>Nº of Steps</i>) visíveis no Wave4IOPT tem que ser maior ou igual a 1.</p> <p>Valor actual no passo, tem que ser superior ao valor mínimo permitido para essa Forma de Onda onde se faz o clique.</p> <p>Caso exista um erro no valor actual no passo (assinalado a cor vermelha no Wave4IOPT), a anterior pré-condição não terá que se verificar.</p> <p>No caso de ser uma <i>Forma de Onda no modo Composto Simples</i>, o clique com o botão esquerdo do rato terá que ser efectuado na linha horizontal de baixo dessa Forma de Onda.</p> <p>No caso de ser um <i>Forma de Onda no modo Composto por Níveis</i>, o clique será com o botão direito do rato, de modo a decrementar o valor no passo. Essa acção só terá efeito caso não haja em memória uma cópia de uma Forma de Onda.</p>
Pós-condições	A Forma de Onda de um evento ou sinal de entrada, nesse passo (<i>step</i>), passará a ter o valor decrementado em uma unidade, em relação ao que tinha anteriormente.

	No caso de ser uma <i>Forma de Onda no modo Bit a Bit</i> , o valor do passo será decrementado em 2^{Bit} , sendo <i>Bit</i> o nº do Bit dessa Forma de Onda.
Sumário	Utilizador faz um clique, com o botão esquerdo do rato (ou com o botão direito do rato no caso de ser um <i>Forma de Onda no modo Compacto por Níveis</i>), num determinado passo (<i>step</i>), em cima da linha de uma Forma de Onda de um evento ou sinal de entrada.
Actor	Utilizador.
Cenário principal	1. Clicar com o botão esquerdo do rato (ou com o botão direito do rato no caso de ser uma <i>Forma de Onda no modo Compacto por Níveis</i>), num determinado passo (<i>step</i>), em cima da linha de um Forma de Onda de um evento ou sinal de entrada, quando o icon do rato fôr uma mão.
Cenários alternativos	Operação de decrementar é precedida de um “Copiar valores dos passos”: 1. a) Caso, antes deste caso de uso, tenha sido realizado com sucesso o caso de uso “Copiar valores dos passos”, a operação de decrementar de uma <i>Forma de Onda no modo Compacto por Níveis</i> será cancelada. Deste modo, após o clique com o botão direito do rato em cima de um passo desta Forma de Onda, irá aparecer o Menu das operações Copiar e Colar (<i>Menu Copy Paste</i>).

Tabela 3.5 – Descrição de Caso de Uso Decrementar com CTRL.

Título do Caso de Uso	Decrementar com CTRL.
Pré-condições	Tem que existir pelo menos uma Forma de Onda de um

	<p>evento ou sinal de entrada visível no Wave4IOPT.</p> <p>Número de Passos (<i>Nº of Steps</i>) visíveis no Wave4IOPT tem que ser maior ou igual a 1.</p> <p>Valor actual no passo, tem que ser superior ao valor mínimo permitido para essa Forma de Onda onde se faz o clique.</p> <p>Caso exista um erro no valor actual no passo (assinalado a cor vermelha no Wave4IOPT), a anterior pré-condição não terá que se verificar.</p> <p>No caso de ser uma <i>Forma de Onda no modo Composto Simples</i>, o clique com o botão esquerdo do rato terá que ser efectuado na linha horizontal de baixo dessa Forma de Onda.</p> <p>No caso de ser uma <i>Forma de Onda no modo Composto por Níveis</i>, o clique será com o botão direito do rato, de modo a decrementar o valor no passo. Essa acção só terá efeito caso não haja em memória uma cópia de uma Forma de Onda.</p>
Pós-condições	<p>A Forma de Onda de um evento ou sinal de entrada, nesse passo (<i>step</i>), passará a ter o valor mínimo permitido para essa Forma de Onda.</p> <p>Os passos (<i>steps</i>) seguintes dessa Forma de Onda terão igualmente esse valor mínimo, até que dois passos consecutivos apresentem, respectivamente, valor mínimo permitido para essa Forma de Onda e valor maior que o mínimo permitido para essa Forma de Onda.</p>
Sumário	<p>Utilizador carrega na tecla CTRL do teclado e faz um clique, com o botão esquerdo do rato (ou com o botão direito do rato no caso de ser uma <i>Forma de Onda no modo Compacto por Níveis</i>), num determinado passo (<i>step</i>), em</p>

	cima da linha de uma Forma de Onda de um evento ou sinal de entrada.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. Carregar na tecla CTRL do teclado e fazer um clique com o botão esquerdo do rato (ou com o botão direito do rato no caso de ser uma <i>Forma de Onda no modo Compacto por Níveis</i>). O clique é efectuado num determinado passo (<i>step</i>), em cima da linha de uma Forma de Onda de um evento ou sinal de entrada, quando o icon do rato fôr uma mão.
Cenários alternativos	<p>Operação não é permitida numa <i>Forma de Onda no modo Bit a Bit</i>:</p> <ol style="list-style-type: none"> 1. a) Caso o clique tenha sido numa das Formas de Onda que constituem uma <i>Forma de Onda no modo Bit a Bit</i>, a operação decrementar com CTRL não será permitida para este sinal. Será apresentada uma mensagem no Wave4IOPT, informando o Utilizador do descrito anteriormente. <p>Operação de decrementar com CTRL é precedida de um “Copiar valores dos passos”:</p> <ol style="list-style-type: none"> 1. b) Caso, antes deste caso de uso, tenha sido realizado com sucesso o caso de uso “Copiar valores dos passos”, e caso o clique (com o botão direito do rato) tenha sido numa <i>Forma de Onda no modo Compacto por Níveis</i>, a operação de decrementar com CTRL uma <i>Forma de Onda no modo Compacto por Níveis</i> será cancelada. Deste modo, após o clique com o botão direito do rato em

	cima de um passo desta Forma de Onda de um evento ou sinal de entrada, irá aparecer o Menu das operações Copiar e Colar (<i>Menu Copy Paste</i>).
--	-----------------------------------------------------------------------------------------------------------------------------------------------------

Tabela 3.6 – Descrição de Caso de Uso Decrementar com CTRL + ALT.

Título do Caso de Uso	Decrementar com CTRL + ALT.
Pré-condições	<p>Tem que existir pelo menos uma Forma de Onda de um evento ou sinal de entrada visível no Wave4IOPT.</p> <p>Número de Passos (<i>Nº of Steps</i>) visíveis no Wave4IOPT tem que ser maior ou igual a 1.</p> <p>Valor actual no passo, tem que ser superior ao valor mínimo permitido para essa Forma de Onda onde se faz o clique.</p> <p>Caso exista um erro no valor actual no passo (assinalado a cor vermelha no Wave4IOPT), a anterior pré-condição não terá que se verificar.</p> <p>No caso de ser uma <i>Forma de Onda no modo Composto Simples</i>, o clique com o botão esquerdo do rato terá que ser efectuado na linha horizontal de baixo desse Forma de Onda.</p> <p>No caso de ser uma <i>Forma de Onda no modo Composto por Níveis</i>, o clique será com o botão direito do rato, de modo a decrementar o valor no passo. Essa acção só terá efeito caso não haja em memória uma cópia de um Forma de Onda.</p>
Pós-condições	<p>A Forma de Onda de um evento ou sinal de entrada, nesse passo (<i>step</i>), passará a ter o valor mínimo permitido para essa Forma de Onda.</p> <p>Todos os passos (<i>steps</i>) seguintes dessa Forma de Onda</p>

	terão igualmente esse valor mínimo.
Sumário	Utilizador carrega nas teclas CTRL e ALT do teclado e faz um clique, com o botão esquerdo do rato (ou com o botão direito do rato no caso de ser uma <i>Forma de Onda no modo Compacto por Níveis</i>), num determinado passo (<i>step</i>), em cima da linha de uma Forma de Onda.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. Carregar nas teclas CTRL e ALT do teclado e fazer um clique com o botão esquerdo do rato (ou com o botão direito do rato no caso de ser uma <i>Forma de Onda no modo Compacto por Níveis</i>). O clique é efectuado num determinado passo (<i>step</i>), em cima da linha dessa Forma de Onda, quando o icon do rato fôr uma mão.
Cenários alternativos	<p>Operação não é permitida numa <i>Forma de Onda no modo Bit a Bit</i>:</p> <ol style="list-style-type: none"> 2. a) Caso o clique tenha sido numa das Formas de Onda que constituem uma <i>Forma de Onda no modo Bit a Bit</i>, a operação decrementar com CTRL + ALT não será permitida para este modo de Forma de Onda. Será apresentada uma mensagem no Wave4IOPT, informando o Utilizador do descrito anteriormente. <p>Operação de decrementar com CTRL + ALT é precedida de um “Copiar valores dos passos”:</p> <ol style="list-style-type: none"> 1. b) Caso, antes deste caso de uso, tenha sido realizado com sucesso o caso de uso “Copiar valores dos passos”, e caso o clique (com o botão direito do rato) tenha sido numa <i>Forma de Onda no</i>

	<p><i>modo Compacto por Níveis</i>, a operação de decrementar com CTRL + ALT será cancelada. Deste modo, após o clique com o botão direito do rato em cima de um passo desta <i>Forma de Onda no modo Compacto por Níveis</i>, irá aparecer o Menu das operações Copiar e Colar (<i>Menu Copy Paste</i>).</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabela 3.7 - Descrição de Caso de Uso Copiar valores dos passos.

Título do Caso de Uso	Copiar valores dos passos.
Pré-condições	<p>Tem que existir pelo menos uma Forma de Onda visível no Wave4IOPT.</p> <p>Número de Passos (<i>Nº of Steps</i>) visíveis no Wave4IOPT tem que ser maior ou igual a 1.</p>
Pós-condições	<p>Ficará guardado em memória o comportamento da Forma de Onda relativo aos passos (<i>steps</i>) seleccionados.</p> <p>Os passos copiados ficarão assinalados no Wave4IOPT com uma animação.</p>
Sumário	O Utilizador desenha um rectângulo, abrangendo os passos da Forma de Onda que deseja copiar. Seguidamente, o Utilizador carrega na opção “Copy” no menu que aparecerá automaticamente.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. Premir o botão esquerdo do rato de forma a desenhar um rectângulo em que abranja os passos de apenas uma Forma de Onda que deseja copiar. 2. Premir a opção “Copy” no menu que aparecerá automaticamente.

Cenários alternativos	<p>Rectângulo desenhado não é válido:</p> <ol style="list-style-type: none"> a) A operação Copiar é interrompida e cancelada porque o rectângulo, desenhado pelo Utilizador, é inválido.
-----------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabela 3.8 - Descrição de Caso de Uso Colar valores dos passos.

Título do Caso de Uso	Colar valores dos passos.
Pré-condições	<p>Tem que existir pelo menos uma Forma de Onda de um evento ou sinal de entrada visível no Wave4IOPT.</p> <p>Número de Passos (<i>Nº of Steps</i>) visíveis no Wave4IOPT tem que ser maior ou igual a 1.</p> <p>Tem que ter sido realizado com sucesso o caso de uso “Copiar valores dos passos”.</p>
Pós-condições	<p>A Forma de Onda de um evento ou sinal de entrada onde se faz a operação Colar, irá apresentar o comportamento da onda, anteriormente copiada, a partir do passo (step) em que se clicou com o botão direito do rato.</p> <p>Os passos (steps) copiados anteriormente, deixarão de estar em memória, bem como, animados no Wave4IOPT.</p>
Sumário	O Utilizador carrega no botão direito do rato em cima da linha de uma Forma de Onda de um evento ou sinal de entrada, no passo (step) a partir do qual deseja efectuar a operação de Colar. Seguidamente, o Utilizador carrega na opção “Paste” no menu que aparecerá automaticamente.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> Premir o botão direito do rato em cima da linha de uma Forma de Onda de um evento ou sinal de

	<p>entrada, no passo (step) a partir do qual deseja efectuar a operação de Colar.</p> <p>2. Premir a opção “Paste” no menu que aparecerá automaticamente. A operação é realizada com sucesso, ficando essa Forma de Onda com o comportamento da onda, copiado anteriormente, a partir do passo (step) em que se clicou com o botão direito do rato.</p>
Cenários alternativos	<p>Operação “Paste” realizada sem sucesso:</p> <p>2. a) A operação não terá sucesso porque a forma de onda, copiada anteriormente, apresenta valores fora dos limites mínimo e máximo que a Forma de Onda de um evento ou sinal de entrada de destino do “Paste” pode tomar. Será apresentada uma mensagem no Wave4IOPT, informando o Utilizador do descrito anteriormente.</p>

Tabela 3.9 – Descrição de Caso de Uso Alterar nome de Forma de Onda.

Título do Caso de Uso	Alterar nome da Forma de Onda.
Pré-condições	Tem que existir pelo menos uma Forma de Onda visível no Wave4IOPT.
Pós-condições	O nome da Forma de Onda será alterado para o nome que o Utilizador escreveu.
Sumário	O Utilizador, através do Menu de uma Forma de Onda, altera o nome da Forma de Onda em questão.
Actor	Utilizador.
Cenário principal	<p>1. Premir o botão direito do rato em cima do nome de</p>

	<p>um Forma de Onda, de modo a aparecer o menu desse Forma de Onda.</p> <p>2. Premir a opção “<i>Rename Waveform</i>” no menu desse Forma de Onda, de modo a aparecer uma caixa de texto, onde o Utilizador irá escrever o novo nome do Forma de Onda.</p> <p>3. Escrever, na caixa de texto, o novo nome do Forma de Onda ou deixar o nome original do Forma de Onda e carregar no botão “OK” da caixa de texto.</p>
Cenários alternativos	<p>Cancelar a operação de alterar o nome do Forma de Onda:</p> <p>3. a) Carregar no botão Cancel ou no X da caixa de texto. A caixa de texto e o menu do Forma de Onda irão desaparecer, cancelando-se assim a operação de alterar o nome do Forma de Onda.</p>

Tabela 3.10 – Descrição de Caso de Uso Corrigir erros das Formas de Onda.

Título do Caso de Uso	Corrigir erros das Formas de Onda.
Pré-condições	Um ou mais Formas de Onda apresentam valores inválidos (nos passos), ou seja, inferiores ao mínimo e/ou superiores ao máximo permitidos para essas Formas de Onda. Passos (steps) com valores inválidos, são assinalados com valores e linhas da cor vermelha.
Pós-condições	<p>Os valores e as linhas referentes aos passos inválidos, deixarão de ter cor vermelha e passarão a cor castanha.</p> <p>Os valores dos passos (anteriormente inválidos) passarão para os limites mínimo ou máximo permitidos para essa Forma de Onda, conforme estivessem abaixo do mínimo ou acima do máximo, respectivamente.</p>

Sumário	O Utilizador irá carregar num botão que irá corrigir todos os valores inválidos dos passos das Formas de Onda, ou seja, corrigir todos os valores que estão fora dos limites mínimo e máximo permitidos para cada Forma de Onda.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. Clicar (com o botão esquerdo do rato) no botão “Fix Errors” do menu Opções do Wave4IOPT. 2. Ao aparecer uma caixa com uma mensagem para confirmar a operação, carregar no botão “OK”.
Cenários alternativos	<p>Cancelar a operação de correcção de erros das Formas de Onda:</p> <ol style="list-style-type: none"> 2. a) Ao aparecer uma caixa com uma mensagem para o Utilizador confirmar a operação, carrega-se no botão Cancel ou no X da caixa de confirmação. A caixa de texto irá desaparecer, cancelando-se assim a operação de correcção de erros das Formas de Onda.

Tabela 3.11 – Descrição de Caso de Uso Alterar valor no passo.

Título do Caso de Uso	Alterar valor no passo.
Pré-condições	Tem que existir pelo menos uma Forma de Onda de um evento ou sinal de entrada visível no Wave4IOPT.
Pós-condições	O valor de um passo (step) é alterado para outro valor ou manter-se-á o mesmo.
Sumário	O Utilizador altera o valor de um passo (step) de uma determinada Forma de Onda de um evento ou sinal de entrada, através de uma opção do menu desse passo.
Actor	Utilizador.

Cenário principal	<ol style="list-style-type: none"> 1. Premir o botão direito do rato em cima do valor de um passo (step) de uma determinada Forma de Onda de um evento ou sinal de entrada, de modo a aparecer o menu desse passo. 2. No menu desse passo, premir a opção “<i>Change Value</i>”, de modo a aparecer uma caixa de texto, onde o Utilizador irá escrever o novo valor do passo. 3. Escrever, na caixa de texto, um novo valor válido para esse passo ou deixar o valor original dessa Forma de Onda e carregar no botão “<i>OK</i>” da caixa de texto.
Cenários alternativos	<p>Cancelar a operação de alterar o valor do passo:</p> <ol style="list-style-type: none"> 3. a) Carregar no botão Cancel ou no X da caixa de texto. A caixa de texto e o menu da Forma de Onda de um evento ou sinal de entrada irão desaparecer, cancelando-se assim a operação de alterar o valor do passo. <p>Novo valor do passo é inválido:</p> <ol style="list-style-type: none"> 3. b) Volta a repetir o ponto 3. do cenário principal, até que seja válido o novo valor do passo.

Tabela 3.12 - Descrição de Caso de Uso Alterar a Base Numérica de uma Forma de Onda.

Título	Alterar a Base Numérica de uma Forma de Onda.
Pré-condições	No Wave4IOPT, tem que estar pelo menos uma Forma de Onda no modo Compacto Simples, Compacto por Níveis ou Bit a Bit.
Pós-condições	Os valores dessa Forma de Onda serão representados na

	forma decimal ou binária ou hexadecimal ou octal, consoante a opção escolhida pelo Utilizador.
Sumário	O Utilizador irá alterar a base numérica (decimal, binária, hexadecimal, octal) dos valores de todos os passos de uma Forma de Onda, mostrados no Wave4IOPT.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega com o botão direito do rato em cima de qualquer valor, de qualquer passo, de uma Forma de Onda de forma a aparecer o Menu <i>Numerical Bases</i>. 2. Neste Menu, estando já seleccionada uma das quatro opções de base numérica (por omissão será a opção decimal), o Utilizador irá alterar a base numérica seleccionando uma das três opções de bases numéricas disponíveis, clicando no botão correspondente. 3. O Menu <i>Numerical Bases</i> desaparece e todos os valores dos passos desse Forma de Onda serão apresentados com a base numérica seleccionada, anteriormente, pelo Utilizador.
Cenário alternativo	<ol style="list-style-type: none"> 2. a) O Utilizador clica fora das três opções (três botões) de base numérica disponíveis do Menu <i>Numerical Bases</i>, exceptuando o botão “<i>Change Value</i>”. 3. a) O Menu desaparece e todos os valores dos passos dessa Forma de Onda irão manter-se representados com a mesma base numérica que tinham no início deste Caso de Uso.

Tabela 3.13 – Descrição de Caso de Uso Recuar uma Acção.

Título	Recuar uma Acção.
Pré-condições	No Wave4IOPT, tem que estar pelo menos uma Forma de Onda de um evento ou sinal de entrada. O Utilizador terá que já ter realizado pelo menos uma acção de edição nessa Forma de Onda.
Pós-condições	O Wave4IOPT apresentará o estado de edição anterior ao actual estado de edição.
Sumário	O Utilizador carrega no botão “Undo Action” (ou a combinação CTRL + Z), de modo a recuar uma acção de edição do Wave4IOPT.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT. 2. Neste painel, carregar uma vez no botão “Undo Action”. 3. Após o clique, o Wave4IOPT apresentará o estado de edição imediatamente anterior àquele em que se encontrava.
Cenário alternativo	<ol style="list-style-type: none"> 1. a) O Utilizador carrega nas teclas “CTRL” e “Z” do teclado, de forma a concretizar a mesma pós-condição deste Caso de Uso.

Tabela 3.14 – Descrição de Caso de Uso Avançar uma Acção.

Título	Avançar uma Acção.
Pré-condições	No Wave4IOPT, tem que estar pelo menos uma Forma de Onda de um evento ou sinal de entrada. O Utilizador terá que já ter realizado com sucesso o Caso de Uso “ <i>Recuar uma Acção</i> ”.
Pós-condições	O Wave4IOPT apresentará o estado de edição posterior ao actual estado de edição.
Sumário	O Utilizador carrega no botão “ <i>Redo Action</i> ” (ou a combinação <i>CTRL + Z</i>), de modo a avançar uma acção de edição do Wave4IOPT.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT. 2. Neste painel, carregar uma vez no botão “<i>Redo Action</i>”. 3. Após o clique, o Wave4IOPT apresentará o estado de edição imediatamente posterior àquele em que se encontrava.
Cenário alternativo	<ol style="list-style-type: none"> 2. a) O Utilizador carrega nas teclas “<i>CTRL</i>” e “<i>Y</i>” do teclado, de forma a concretizar a mesma pós-condição deste Caso de Uso.

Tabela 3.15 – Descrição de Caso de Uso Ampliar.

Título	Ampliar.
Pré-condições	A percentagem de ampliação terá que ser inferior a 175%, de

	forma a ser visível o efeito da ampliação.
Pós-condições	No Wave4IOPT, o tamanho de todas as Formas de Onda visíveis, de seus nomes e valores, será aumentado.
Sumário	O Utilizador carrega no botão “+” no painel lateral de opções do Wave4IOPT, de forma a aumentar o tamanho de todas as Formas de Onda visíveis e seus correspondentes nomes e valores.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT. 2. Neste painel, carregar uma vez no botão “+”. 3. Após o clique, todas as formas de onda visíveis, bem como, os nomes das Formas de Onda e os valores de cada passo serão ampliados, aumentando de tamanho 10% ou 25% (conforme o caso).
Cenário alternativo	

Tabela 3.16 – Descrição de Caso de Uso Reduzir.

Título	Reduzir.
Pré-condições	A percentagem de ampliação (mostrada no painel lateral de opções do Wave4IOPT) terá que ser superior a 25%, de forma a ser visível o efeito da redução do tamanho.
Pós-condições	No Wave4IOPT, o tamanho de todas as Formas de Onda visíveis, de seus nomes e valores, será diminuído.
Sumário	O Utilizador carrega no botão “-” no painel lateral de opções do Wave4IOPT, de forma a diminuir o tamanho de todas as

	Formas de Onda visíveis e seus correspondentes nomes e valores.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT. 2. Neste painel, carregar uma vez no botão “-”. 3. Após o clique, todas as formas de onda visíveis, bem como, os nomes das Formas de Onda e os valores de cada passo serão reduzidos, diminuindo de tamanho 10% ou 25% (conforme o caso).
Cenário alternativo	

Tabela 3.17 – Descrição de Caso de Uso Alterar cor do nome da Forma de Onda.

Título	Alterar cor do nome da Forma de Onda.
Pré-condições	
Pós-condições	A cor do nome de um dos sub-tipos de Forma de Onda (Sinais / Eventos / Lugares / Transições), será alterada, dependendo da caixa de edição de cor a que se aceda.
Sumário	O Utilizador acede a uma caixa de edição de cor, correspondente à cor do nome de um dos sub-tipos de Forma de Onda, afim de alterar essa cor.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT.

	<ol style="list-style-type: none"> 2. Neste painel, carregar uma vez num dos quatro botões coloridos (<i>“Signals”</i> / <i>“Events”</i> / <i>“Places”</i> / <i>“Transitions”</i>), consoante se queira alterar a cor dos nomes do sub-tipo de Forma de Onda referido. 3. Após o clique, irá aparecer uma caixa de escolha de cor, e o Utilizador selecciona a cor que quer e clica no botão <i>“OK”</i> de forma a confirmar a selecção. 4. Após a confirmação, todos os nomes do sub-tipo de Forma de Onda escolhido anteriormente, terão a cor seleccionada pelo Utilizador no ponto anterior.
Cenário alternativo	

Tabela 3.18 – Descrição de Caso de Uso Mostrar Forma de Onda.

Título	Mostrar Forma de Onda.
Pré-condições	Existir um ficheiro JSON já carregado pelo Wave4IOPT, com pelo menos uma Forma de Onda na sua estrutura, mesmo que, de momento, essa Forma de Onda não esteja a ser visualizada no Wave4IOPT.
Pós-condições	A Forma de Onda ficará visível no Wave4IOPT.
Sumário	O Utilizador torna uma Forma de Onda visível, podendo utilizar duas formas: ou a partir do painel de topo <i>“Choose Waveforms”</i> onde pode seleccionar a Forma de Onda individualmente, ou a partir das caixas de selecção, na secção <i>“Show”</i> no painel lateral de opções do Wave4IOPT, onde pode seleccionar a Forma de Onda pretendida escolhendo o sub-tipo e tipo ao qual pertence.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do

	<p>Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT.</p> <ol style="list-style-type: none"> 2. Neste painel, seleccionar a caixa de selecção <i>“Choose Waveforms from list”</i>, de forma a que o botão <i>“Choose Waveforms”</i> fique disponível para ser premido. 3. Premir o botão <i>“Choose Waveforms”</i>, de forma a que possa a ser visível o painel de topo <i>“Choose Waveforms”</i>. 4. No painel de topo, seleccionar da lista <i>“Available Waveforms Not Added”</i> a Forma de Onda que se pretende visualizar no Wave4IOPT e premir o botão <i>></i>, de forma a que o Forma de Onda passe para a lista <i>“Waveforms Added”</i>. 5. Clicar no botão <i>“Confirm”</i> no painel de topo de forma a que a Forma de Onda escolhida seja visualizada no Wave4IOPT.
Cenário alternativo	<ol style="list-style-type: none"> 2. a) Neste painel, na secção <i>“Show”</i>, seleccionar as caixas de selecção que forem necessárias, de forma a que a Forma de Onda que se pretende visualizar possa aparecer no Wave4IOPT. Caso as caixas que pretendia seleccionar já estavam seleccionadas e a Forma de Onda (que pretende visualizar no Wave4IOPT) não esteja a aparecer, terá que desseleccionar essas caixas de selecção e voltar a selecciona-las.

Tabela 3.19 – Descrição de Caso de Uso Esconder Forma de Onda.

Título	Esconder Forma de Onda.
Pré-condições	Existir um ficheiro JSON já carregado pelo Wave4IOPT, com

	pelo menos uma Forma de Onda na sua estrutura, estando de momento visível no Wave4IOPT.
Pós-condições	A Forma de Onda ficará não visível no Wave4IOPT.
Sumário	<p>O Utilizador torna uma Forma de Onda não visível no Wave4IOPT, podendo utilizar três formas:</p> <ul style="list-style-type: none"> - a partir do painel de topo “Choose Waveforms” onde pode seleccionar individualmente a Forma de Onda a esconder; - a partir das caixas de selecção na secção “Show” (no painel lateral de opções do Wave4IOPT) onde pode esconder a Forma de Onda desseleccionando as caixas do sub-tipo e tipo ao qual a Forma de Onda pertence; - a partir do Menu Individual da Forma de Onda na opção “Do Not Show Waveform”.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT. 2. Neste painel, seleccionar a caixa de selecção “Choose Waveforms from list”, de forma a que o botão “Choose Waveforms” fique disponível para ser premido. 3. Premir o botão “Choose Waveforms”, de forma a que possa a ser visível o painel de topo “Choose Waveforms”. 4. No painel de topo, seleccionar a partir da lista “Waveforms Added” a Forma de Onda que se pretende não visualizar no Wave4IOPT e premir o botão < de forma a que essa Forma de Onda passe para a lista “Available Waveforms Not Added”.

	<p>5. Clicar no botão “Confirm” no painel de topo de forma a que a Forma de Onda escolhida não seja visualizada no Wave4IOPT.</p>
Cenário alternativo	<p>Esconder a Forma de Onda a partir das caixas de selecção na secção Show do painel lateral de opções do Wave4IOPT:</p> <p>3. a) Neste painel, na secção “Show”, desseleccionar as caixas de selecção que forem necessárias, de forma a que a Forma de Onda não seja visualizada no Wave4IOPT. Caso as caixas que pretendia desseleccionar já estavam desseleccionadas e a Forma de Onda que pretende não visualizar no Wave4IOPT esteja a aparecer, terá que seleccionar essas caixas de selecção e voltar a desselecciona-las.</p> <p>Esconder a Forma de Onda a partir do Menu Individual da Forma de Onda, na opção “Do Not Show Waveform”:</p> <p>1. O Utilizador carrega com o botão direito do rato em cima do nome da Forma de Onda que deseja esconder, de modo a aparecer o Menu Individual da Forma de Onda.</p> <p>2. Nesse Menu Individual da Forma de Onda, seleccionar a opção “Do Not Show Waveform”, de forma a não visualizar essa Forma de Onda no Wave4IOPT.</p>

Tabela 3.20 – Descrição de Caso de Uso Alterar modo de visualização da Forma de Onda.

Título	Alterar modo de visualização da Forma de Onda.
Pré-condições	Existir um ficheiro JSON já carregado pelo Wave4IOPT, com pelo menos uma Forma de Onda no modo Compacto

	Simples, Compacto por Níveis ou Bit a Bit na sua estrutura, estando visível no Wave4IOPT.
Pós-condições	A Forma de Onda será apresentada num modo diferente daquele que apresentado anteriormente no Wave4IOPT.
Sumário	O Utilizador acede ao Menu Individual da Forma de Onda de forma a modificar o modo de visualização dessa Forma de Onda no Wave4IOPT.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega com o botão direito do rato em cima do nome da Forma de Onda, de forma a aparecer o seu Menu Individual. 2. Nesse Menu Individual, colocar o cursor do rato em cima da opção “<i>Waveform view mode</i>”, de forma a aparecer um sub-menu com os três possíveis modos de visualização dessa Forma de Onda. 3. Nesse sub-menu, premir uma das duas opções disponíveis, que não apresentam o símbolo \surd, de forma a alterar o modo de visualização dessa Forma de Onda para o modo de visualização pretendido.
Cenário alternativo	<p>Utilizador carrega na opção “<i>Waveform view mode</i>”, anulando a alteração do modo de visualização da Forma de Onda:</p> <ol style="list-style-type: none"> 2. a) Neste Menu Individual da Forma de Onda, ao carregar com o botão esquerdo do rato na opção “<i>Waveform view mode</i>”, irá desaparecer o Menu Individual, não se alterando assim o modo de visualização da Forma de Onda. <p>Utilizador carrega noutro local que não um dos dois modos disponíveis de visualização, anulando a alteração do modo de</p>

	<p>visualização da Forma de Onda:</p> <p>3. a) Nesse sub-menu, ao clicar com o botão esquerdo do rato na opção que apresenta o símbolo $\sqrt{}$ ou noutra local desse sub-menu que não nos dois botões disponíveis para clicar, quer o sub-menu quer o Menu Individual da Forma de Onda irão desaparecer, não se alterando o modo de visualização da Forma de Onda.</p>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabela 3.21 – Descrição de Caso de Uso Mover Forma de Onda uma posição para cima.

Título	Mover Forma de Onda uma posição para cima.
Pré-condições	Existir um ficheiro JSON já carregado pelo Wave4IOPT, com pelo menos uma Forma de Onda visível no Wave4IOPT ou pelo menos duas Formas de Onda visíveis no Wave4IOPT de forma a ser visível no Wave4IOPT o efeito desta acção.
Pós-condições	A Forma de Onda irá aparecer uma posição acima da ocupada anteriormente, na lista de Formas de Onda visíveis no Wave4IOPT.
Sumário	O Utilizador, a partir do Menu Individual de uma Forma de Onda, selecciona a opção de mover essa Forma de Onda uma posição, no sentido ascendente, na lista de Formas de Onda visíveis no Wave4IOPT.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega com o botão direito do rato em cima do nome de uma Forma de Onda que deseja mover, de forma a aparecer o Menu Individual dessa Forma de Onda. 2. Nesse Menu Individual da Forma de Onda, carregar uma vez na 1ª opção “Move Up”, de forma a mover a

	<p>Forma de Onda para cima.</p> <p>3. A Forma de Onda irá aparecer uma posição acima da ocupada anteriormente, na lista de Formas de Onda visíveis no Wave4IOPT.</p>
Cenário alternativo	<p>A Forma de Onda não move uma posição para cima:</p> <p>3. a) A Forma de Onda não irá aparecer uma posição acima da ocupada anteriormente, por já estar na posição mais acima possível, na lista de Formas de Onda visíveis no Wave4IOPT.</p>

Tabela 3.22 – Descrição de Caso de Uso Mover Forma de Onda uma posição para baixo.

Título	Mover Forma de Onda uma posição para baixo.
Pré-condições	Existir um ficheiro JSON já carregado pelo Wave4IOPT, com pelo menos uma Forma de Onda visível no Wave4IOPT ou pelo menos duas Formas de Onda visíveis no Wave4IOPT de forma a ser visível no Wave4IOPT o efeito desta acção.
Pós-condições	A Forma de Onda irá aparecer uma posição abaixo da ocupada anteriormente, na lista de Formas de Onda visíveis no Wave4IOPT.
Sumário	O Utilizador, a partir do Menu Individual de uma Forma de Onda, selecciona a opção de mover essa Forma de Onda uma posição, no sentido descendente, na lista de Formas de Onda visíveis no Wave4IOPT.
Actor	Utilizador.
Cenário principal	<p>1. O Utilizador carrega com o botão direito do rato em cima do nome de uma Forma de Onda que deseja mover, de forma a aparecer o Menu Individual dessa</p>

	<p>Forma de Onda.</p> <ol style="list-style-type: none"> 2. Nesse menu individual da Forma de Onda, carregar uma vez na 2ª opção “<i>Move Down</i>”, de forma a mover a Forma de Onda para baixo. 3. A Forma de Onda irá aparecer uma posição abaixo da ocupada anteriormente, na lista de Formas de Onda visíveis no Wave4IOPT.
Cenário alternativo	<p>A Forma de Onda não move uma posição para baixo:</p> <ol style="list-style-type: none"> 3. a) A Forma de Onda não irá aparecer uma posição abaixo da ocupada anteriormente, por já estar na posição mais abaixo possível, na lista de Formas de Onda visíveis no Wave4IOPT.

Tabela 3.23 – Descrição de Caso de Uso Apagar Forma de Onda.

Título	Apagar Forma de Onda.
Pré-condições	Existir um ficheiro JSON já carregado pelo Wave4IOPT, com pelo menos uma Forma de Onda visível no Wave4IOPT.
Pós-condições	A Forma de Onda irá desaparecer do Wave4IOPT, deixando de estar também disponível para futura adição ao Wave4IOPT.
Sumário	O Utilizador, a partir do Menu Individual de uma Forma de Onda, selecciona a opção de apagar essa Forma de Onda do Wave4IOPT.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega com o botão direito do rato em cima do nome de uma Forma de Onda, de forma a aparecer o Menu Individual dessa Forma de Onda. 2. Nesse Menu Individual da Forma de Onda, carregar

	na 3ª opção “Delete Waveform”, de forma a apagar essa Forma de Onda do Wave4IOPT.
Cenário alternativo	

Tabela 3.24 – Descrição de Caso de Uso Duplicar Forma de Onda.

Título do Caso de Uso	Duplicar Forma de Onda.
Pré-condições	Existir um ficheiro JSON já carregado pelo Wave4IOPT, com pelo menos uma Forma de Onda visível no Wave4IOPT.
Pós-condições	É criado uma nova Forma de Onda, em tudo igual à Forma de Onda de origem, excepto o nome. Esta nova Forma de Onda será colocada na última posição da lista de Formas de Onda visíveis no Wave4IOPT.
Sumário	O Utilizador, a partir do Menu Individual de uma Forma de Onda, selecciona a opção de duplicar essa Forma de Onda no Wave4IOPT.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega com o botão direito do rato em cima do nome de uma Forma de Onda, de modo a aparecer o Menu Individual dessa Forma de Onda. 2. Nesse Menu Individual da Forma de Onda, carregar na opção “Duplicate Waveform”, de forma a duplicar essa Forma de Onda. 3. Irá aparecer uma caixa de texto, onde o Utilizador irá escrever o nome para essa nova Forma de Onda duplicada e carregar no botão “OK” da caixa de texto. 4. A Forma de Onda duplicada aparecerá na última posição da lista de Formas de Onda visíveis no

	Wave4IOPT e será idêntica à Forma de Onda de origem, excepto no nome.
Cenários alternativos	<p>Cancelar a operação de duplicar Forma de Onda:</p> <p>3. a) Carregar no botão Cancel ou no X da caixa de texto. A caixa de texto e o Menu Individual da Forma de Onda irão desaparecer, cancelando-se assim a operação de duplicar Forma de Onda.</p> <p>Nome para a Forma de Onda duplicada, escolhido pelo Utilizador, já se encontra atribuído:</p> <p>4. b) Irá aparecer uma mensagem de aviso no Wave4IOPT, indicando que o nome escolhido pelo Utilizador, no passo 3., já se encontra atribuído. Pelo que, o Utilizador, terá que escrever novamente, na caixa de texto, um nome que ainda não esteja a ser utilizado por nenhuma das Formas de Onda visíveis ou invisíveis no Wave4IOPT. Este ponto irá repetir-se até que o Utilizador satisfaça essa condição.</p>

Tabela 3.25 – Descrição de Caso de Uso Abrir ficheiro JSON.

Título	Abrir ficheiro JSON.
Pré-condições	Terá que existir, num disco interno ou externo ao Computador do Utilizador, um ficheiro JSON com estrutura igual ao objecto “allWaveforms”.
Pós-condições	O Wave4IOPT ficará com uma estrutura carregada em memória e com uma ou mais Formas de Onda visíveis no Wave4IOPT.
Sumário	O Utilizador, a partir do botão “Open JSON file...” (ou a combinação CTRL + O), acede ao Ambiente de Trabalho (Desktop) do seu Computador, de forma a poder seleccionar e

	abrir um ficheiro JSON, que contém a informação das Formas de Onda.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT. 2. Neste painel, carregar no botão “Open JSON file...”, de forma a abrir uma janela de selecção de ficheiros do Ambiente de Trabalho (<i>Desktop</i>) do Computador do Utilizador. 3. Seleccionar um ficheiro JSON, com estrutura de Formas de Onda idêntica à do objecto “allWaveforms”, e finalizar a operação carregando no botão “Open” da janela. 4. Se o carregamento das Formas de Onda fôr bem sucedido, para além de aparecer uma mensagem com o seguinte texto “Great success! All the File APIs are supported.”, irão também aparecer uma ou mais Formas de Onda visíveis no Wave4IOPT. Se não fôr visível qualquer Forma de Onda, tal poderá não ser indicativo de falha, caso hajam Formas de Onda na coluna “Available Waveforms Not Added” e não haja qualquer Forma de Onda na coluna “Waveforms Added” do painel de topo “Choose Waveforms”. Neste caso as Formas de Onda foram bem carregadas, apenas não estão visíveis no Wave4IOPT.
Cenário alternativo	<p>O carregamento do ficheiro JSON não foi bem sucedido:</p> <ol style="list-style-type: none"> 4. a) Se o carregamento das Formas de Onda não fôr bem sucedido, nenhuma Forma de Onda será visível

	no Wave4IOPT. Poderão aparecer os nomes das Formas de Onda na coluna “ <i>Waveforms Added</i> ” do painel de topo “ <i>Choose Waveforms</i> ”, mas tal significará que existe pelo menos um erro no objecto “ <i>allWaveforms</i> ” que impossibilita a visualização de qualquer Forma de Onda no Wave4IOPT.
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabela 3.26 – Descrição de Caso de Uso Guardar como ficheiro JSON.

Título	Guardar como ficheiro JSON.
Pré-condições	Existir um ficheiro JSON já carregado pelo Wave4IOPT, com pelo menos uma Forma de Onda.
Pós-condições	A informação sobre as Formas de Onda ficará guardada num ficheiro JSON, guardado num disco interno ou externo ao computador do Utilizador.
Sumário	O Utilizador, a partir do botão “ <i>Save as JSON file...</i> ” (ou da combinação CTRL + S), irá guardar a informação sobre as Formas de Onda num ficheiro JSON.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT. 2. Neste painel, carregar no botão “<i>Save as JSON file...</i>”, de forma a abrir, numa nova aba do <i>browser</i>, o texto referente a toda a informação das Formas de Onda. 3. Carregar, com o botão direito do rato, na janela dessa nova aba, de forma a aparecer um menu de contexto. 4. Nesse menu de contexto, seleccionar a opção “<i>Save</i>

	<p><i>as...</i>", de forma a aparecer a janela que possibilita guardar a informação das Formas de Onda num ficheiro.</p> <p>5. Seleccionar a localização pretendida, escrever um nome para esse ficheiro e finalizar a operação carregando no botão "<i>Save</i>".</p>
Cenário alternativo	

Tabela 3.27 – Descrição de Caso de Uso Alterar número de passos visíveis.

Título	Alterar número de passos visíveis.
Pré-condições	<p>Existir um ficheiro JSON já carregado pelo Wave4IOPT.</p> <p>Estar pelo menos uma Forma de Onda visível no Wave4IOPT, de forma a ser visível ao Utilizador o efeito da acção.</p>
Pós-condições	Todos as Formas de Onda visíveis no Wave4IOPT ficarão com o número de passos visíveis escolhido pelo Utilizador.
Sumário	O Utilizador altera o número de passos visíveis, na caixa de texto " <i>Nº Steps</i> ".
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT. 2. Neste painel, carregar na caixa de texto "<i>Nº Steps</i>" e digitar o número de passos que pretende ser visualizado e carregar na tecla "<i>Enter</i>", de forma a finalizar a acção. 3. Todos as Formas de Onda serão apresentadas com o número de passos (<i>steps</i>) que o Utilizador digitou no

	passo 3.
Cenário alternativo	<p>Número de passos, digitado pelo Utilizador, é inválido:</p> <p>3. a) Irá aparecer uma mensagem no Wave4IOPT com o texto <i>"Invalid Number of Steps! Please enter a valid number."</i>, indicando que o Utilizador digitou um número inválido.</p> <p>Considera-se um número inválido caso não seja um número, ou caso o número seja igual ou inferior a zero.</p>

Tabela 3.28 – Descrição de Caso de Uso Esconder todos as Formas de Onda.

Título	Esconder todos as Formas de Onda.
Pré-condições	<p>Existir um ficheiro JSON já carregado pelo Wave4IOPT.</p> <p>Estar pelo menos uma Forma de Onda visível no Wave4IOPT, de modo a ser visível ao Utilizador o efeito da acção.</p>
Pós-condições	<p>Todos as Formas de Onda visíveis no Wave4IOPT deixarão de estar visíveis no Wave4IOPT.</p> <p>No painel de topo <i>"Choose Waveforms"</i>, na coluna <i>"Available Waveforms Not Added"</i> estarão todos os nomes das Formas de Onda não visíveis e na coluna <i>"Waveforms Added"</i> não estará qualquer nome de Forma de Onda.</p>
Sumário	O Utilizador, ao carregar no botão <i>"Clear"</i> (ou na combinação CTRL + E), fará com que desapareçam todos as Formas de Onda anteriormente visíveis no Wave4IOPT.
Actor	Utilizador.
Cenário principal	<p>1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do</p>

	Wave4IOPT. 2. Neste painel, carregar no botão “Clear”, de forma a que todos as Formas de Onda visíveis deixem de estar visíveis.
Cenário alternativo	

Tabela 3.29 – Descrição de Caso de Uso Visualizar nome e número do passo de uma Forma de Onda.

Título	Visualizar nome e número do passo de uma Forma de Onda.
Pré-condições	Existir um ficheiro JSON já carregado pelo Wave4IOPT. Estar pelo menos uma Forma de Onda visível no Wave4IOPT, de modo a ser visível ao Utilizador o efeito da acção.
Pós-condições	
Sumário	O Utilizador ao passar com o cursor do rato por cima da onda de determinada Forma de Onda em determinado passo, poderá observar no painel lateral de Opções do Wave4IOPT o nome e o número do passo dessa Forma de Onda.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega no botão de Opções do Wave4IOPT (no canto superior direito do ecrã), de modo a abrir o painel lateral de opções do Wave4IOPT. 2. O Utilizador deixa o cursor do rato em cima da onda de uma Forma de Onda, num determinado passo. 3. O Utilizador poderá visualizar, no painel lateral de opções do Wave4IOPT, na caixa “Name” e “Step”, respectivamente, o nome da Forma de Onda e o número do passo dessa Forma de Onda.

Cenário alternativo	
---------------------	--

Tabela 3.30 – Descrição de Caso de Uso Modificar Forma de Onda Tudo a 1.

Título do Caso de Uso	Modificar Forma de Onda Tudo a 1.
Pré-condições	<p>Tem que existir pelo menos uma Forma de Onda de um evento ou sinal de entrada visível no Wave4IOPT, pois esta funcionalidade só é válida para essas Formas de Onda.</p> <p>Número de Passos (<i>Nº of Steps</i>) visíveis no Wave4IOPT tem que ser maior ou igual a 1.</p>
Pós-condições	A Forma de Onda de um evento ou sinal de entrada, passará a ter os valores de todos os passos igual a 1.
Sumário	O Utilizador, a partir de um sub-menu do Menu Individual de uma Forma de Onda (evento ou sinal de entrada), selecciona a opção para modificar todos os valores da Forma de Onda para 1.
Actor	Utilizador.
Cenário principal	<ol style="list-style-type: none"> 1. O Utilizador carrega com o botão direito do rato em cima do nome da Forma de Onda, de forma a aparecer o seu Menu Individual. 2. Nesse Menu Individual, colocar o cursor do rato em cima da opção “<i>Modify Waveform to...</i>”, de forma a aparecer um sub-menu com quatro opções de escolha. 3. Nesse sub-menu, premir o 1º botão: “1”.
Cenários alternativos	<p>Modificar Forma de Onda Tudo a 0:</p> <p>Utilizador carrega na opção “0” no sub-menu do <i>Modify Waveform</i>, colocando todos os valores desta Forma de Onda</p>

	<p>a zero:</p> <p>3. a) Nesse sub-menu de <i>Modify Waveform</i>, premir o 2º botão: “0”. Todos os valores desta Forma de Onda ficam com o valor zero.</p> <p>Modificar Forma de Onda <i>Random</i>:</p> <p>Utilizador carrega na opção “<i>Random</i>” no sub-menu do <i>Modify Waveform</i>, ficando esta Forma de Onda com valores aleatórios ao longo do tempo:</p> <p>3. b) Nesse sub-menu de <i>Modify Waveform</i>, premir o 3º botão: “<i>Random</i>”. Assim, cada um dos valores nos passos desta Forma de Onda fica com um valor compreendido entre os valores mínimo e máximo permitidos para essa Forma de Onda.</p> <p>Modificar Forma de Onda $F(x)$:</p> <p>Utilizador carrega na opção “$F(x)$” no sub-menu do <i>Modify Waveform</i>, ficando esta Forma de Onda com valores resultantes dos cálculos da fórmula escolhida pelo Utilizador:</p> <p>3. c) Nesse sub-menu de <i>Modify Waveform</i>, premir o 4º botão: “$F(x)$”.</p> <p>4. c) Surgirá uma caixa de diálogo de modo a que o Utilizador insira uma função $F(x)$, que irá servir para calcular os novos valores desta Forma de Onda.</p> <p>5. c) Após a introdução da fórmula pretendida e carregar no botão “OK”, a fórmula será avaliada e se os novos valores da Forma de Onda são ou não válidos para serem representados no Wave4IOPT, ou</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	seja, se estão compreendidos entre os valores mínimo e máximo (inclusive) permitidos para essa Forma de Onda.
--	---------------------------------------------------------------------------------------------------------------

3.2.3. Diagramas de Actividade

De modo a compreender-se melhor o funcionamento do Wave4IOPT e das suas funcionalidades, apresentam-se de seguida os Diagramas de Actividade de alguns dos Casos de Uso mais importantes:

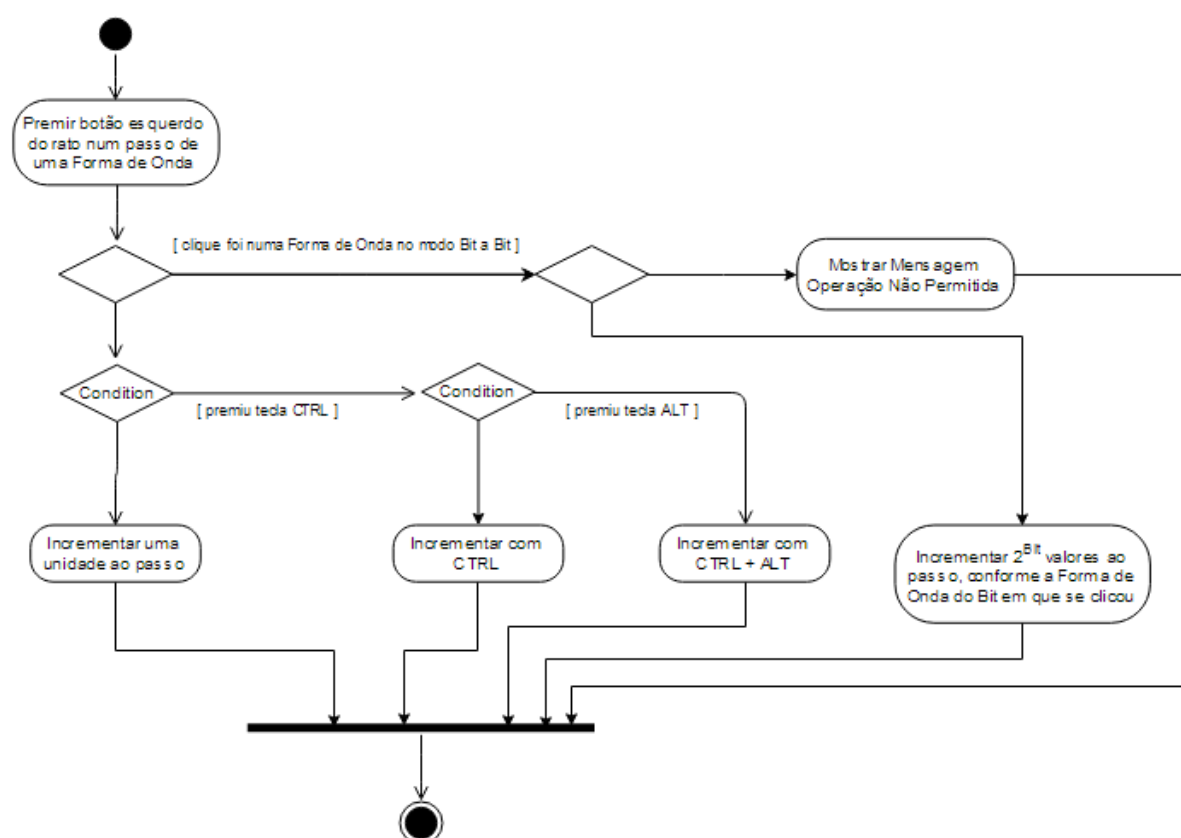


Figura 3.27: Diagrama de Actividade dos Casos de Uso Incrementar valores nos passos, Incrementar com CTRL e Incrementar com CTRL + ALT.

Na **Figura 3.27** estão representados três Casos de Uso, descritos em pormenor nas **Tabelas 3.1, 3.2 e 3.3**, nomeadamente, Incrementar valores nos passos, Incrementar com CTRL e Incrementar com CTRL + ALT.

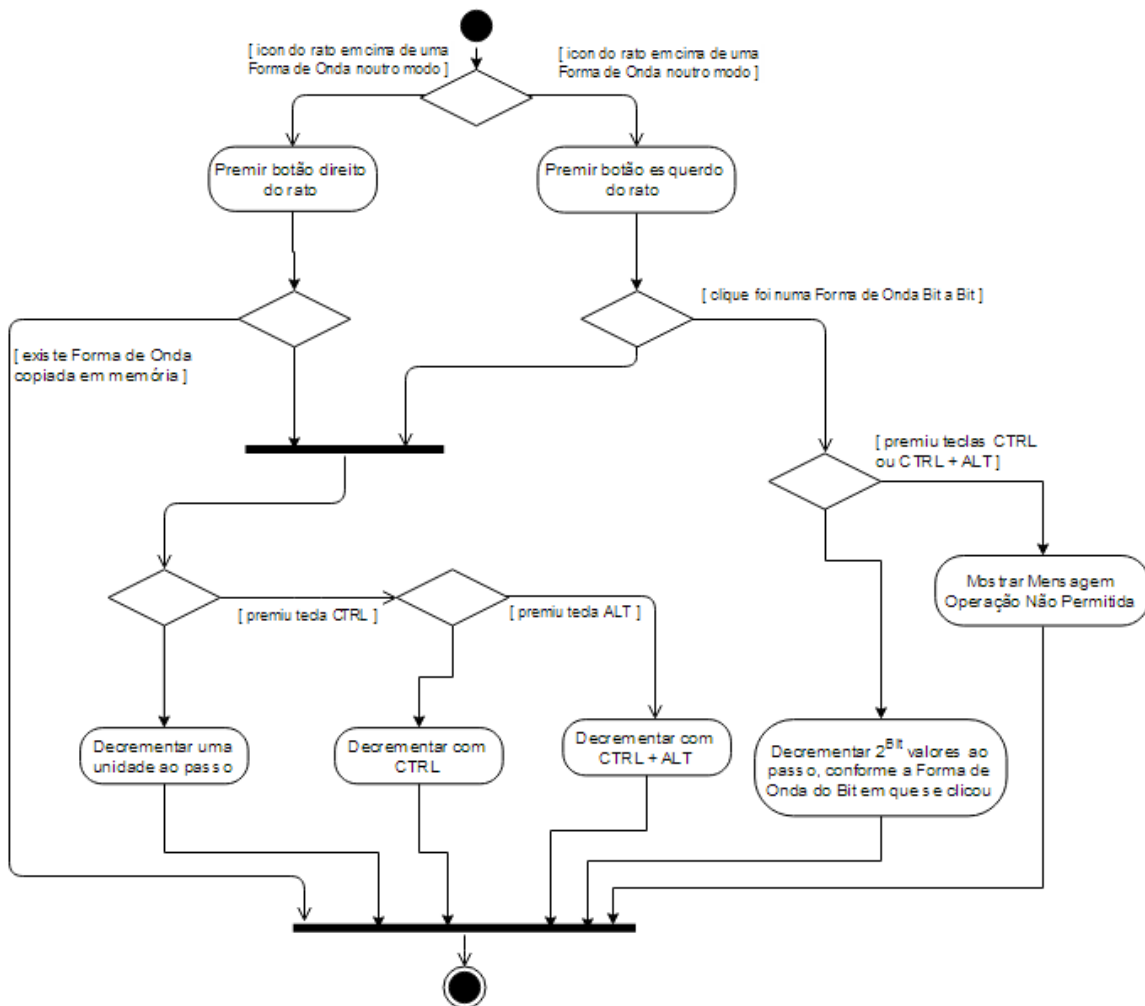


Figura 3.28: Diagrama de Actividade dos Casos de Uso Decrementar valores nos passos, Decrementar com CTRL e Decrementar com CTRL + ALT.

Na **Figura 3.28** estão representados também três Casos de Uso, descritos em pormenor nas **Tabelas 3.4, 3.5 e 3.6**, nomeadamente, “Decrementar valores nos passos”, “Decrementar com CTRL” e “Decrementar com CTRL + ALT”. Realce para o facto de não ser possível decrementar um valor numa Forma de Onda no modo

Compacto por Níveis caso exista alguma Forma de Onda em memória. Isto acontece pois ambas as situações estão ligadas ao pressionar do botão direito em cima da Forma de Onda, existindo assim um conflito, sendo resolvido ao dar a prioridade ao “Colar valores dos passos”.

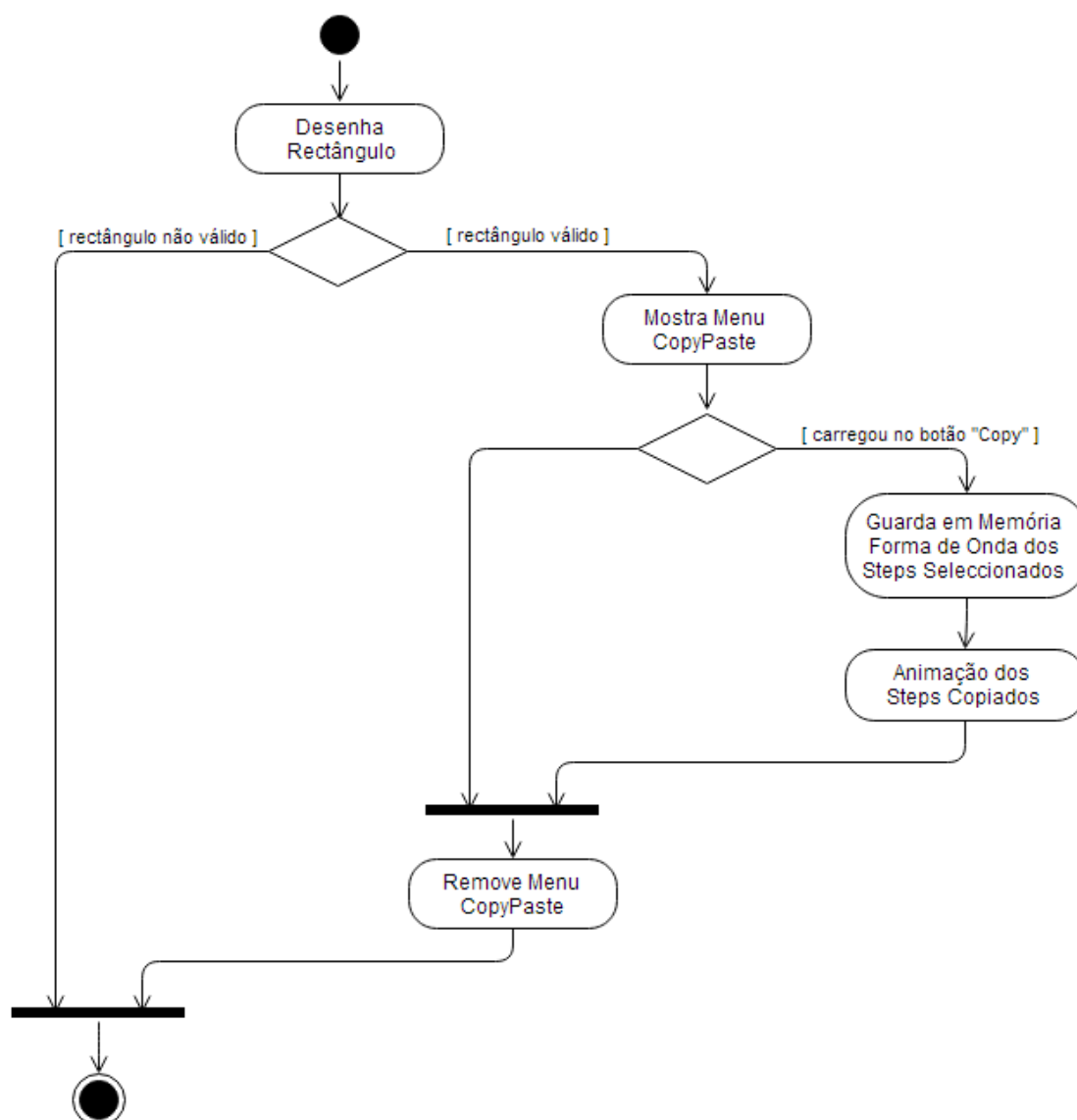


Figura 3.29: Diagrama de Actividade do Casos de Uso Copiar valores dos passos.

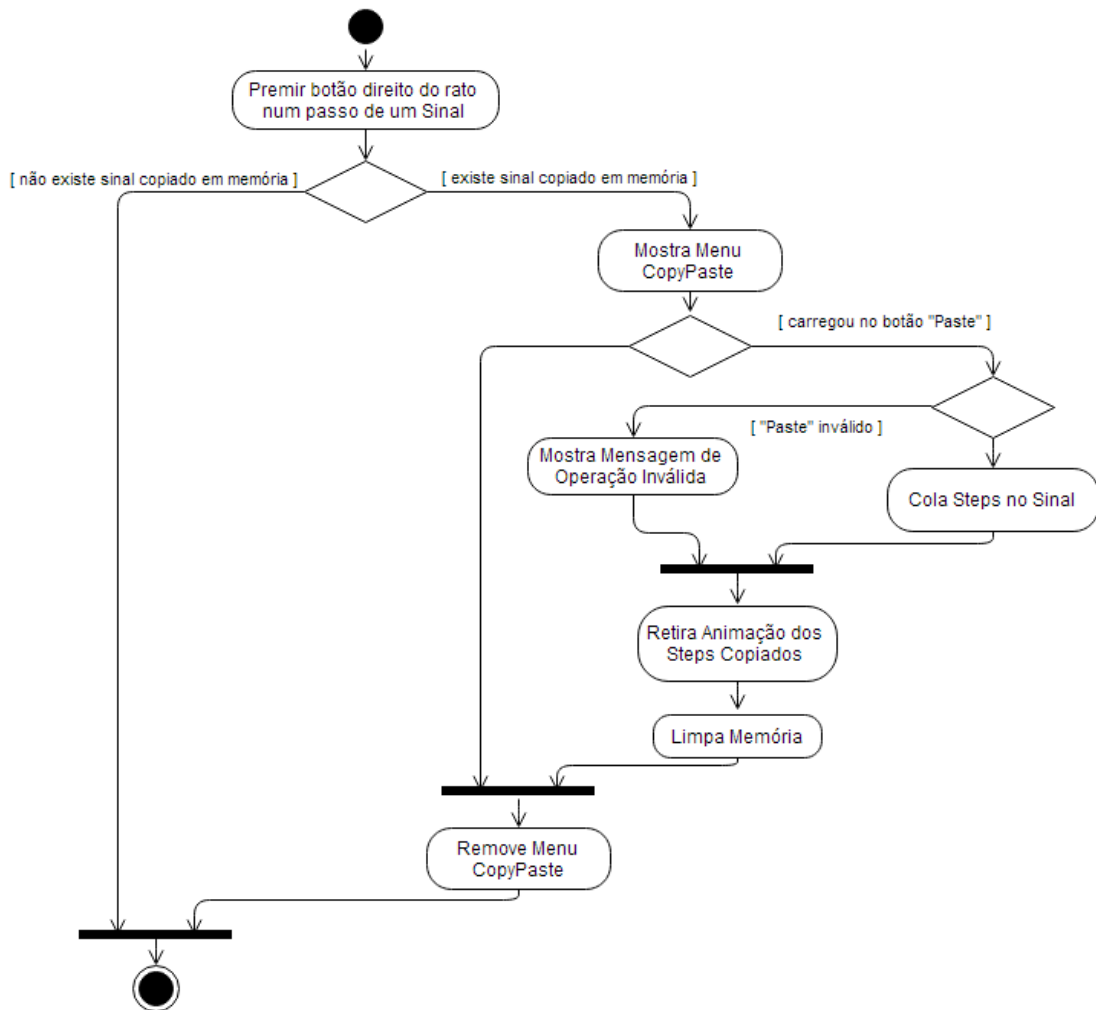


Figura 3.30: Diagrama de Actividade do Caso de Uso Colar valores dos passos.

Nas **Figura 3.29** e **3.30** estão representados dois Casos de Uso, descritos em pormenor nas **Tabelas 3.7** e **3.8**, respectivamente, “Copiar valores dos passos” e “Colar valores dos passos”. De notar que o Copiar/Colar valores dos passos só acontece uma vez. Caso se pretenda copiar os mesmos passos da mesma Forma de Onda, o Utilizador terá que afectar novamente, desde início, o processo de cópia dessa Forma de Onda.



Exemplos Práticos

Neste capítulo, são apresentados exemplos práticos de aplicação da ferramenta Wave4IOPT desenvolvida, utilizando, nomeadamente, dois exemplos de Redes de Petri IOPT que modelam o controlador para um parque de estacionamento: com uma entrada e uma saída e com duas entradas e uma saída.

4.1. Parque de estacionamento com uma entrada e uma saída

Neste exemplo, utilizando uma RdP IOPT disponível na IOPT-Tools, pretende-se modelar um controlador de entradas e saídas de um parque de estacionamento com uma entrada e uma saída com capacidade para 100 veículos (**Figura 4.1**).

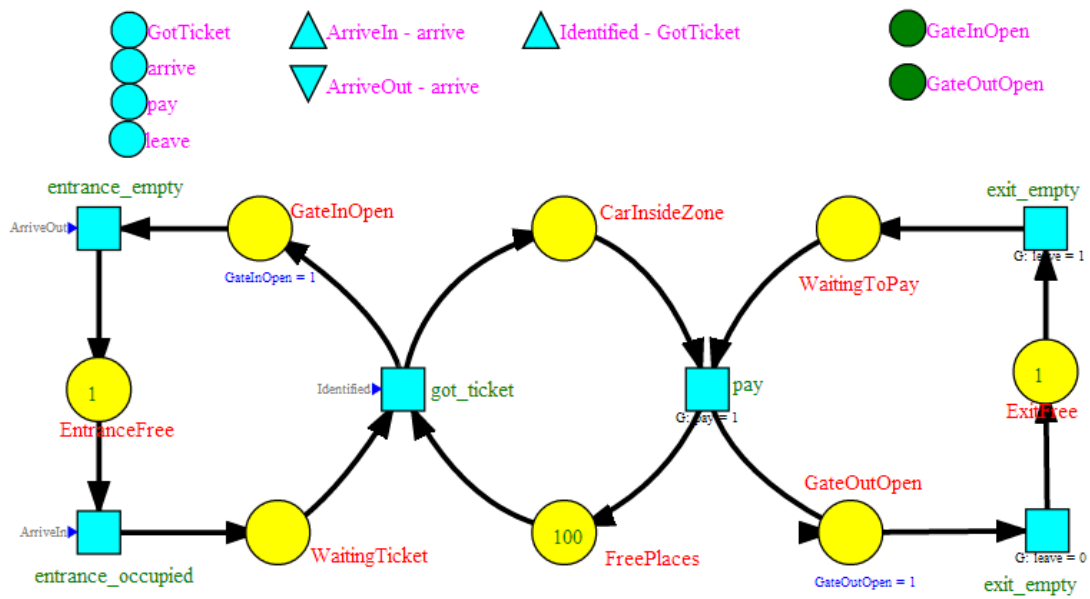


Figura 4.1: RdP IOPT de um controlador de entradas e saídas de um parque de estacionamento com 1 entrada e 1 saída.¹

Observando o modelo, os lugares estão desenhados com círculos amarelos, as transições por rectângulos a azul-claro e os arcos a preto. Os eventos são desenhados como triângulos e os sinais por pequenos círculos. Os eventos e sinais de entrada são apresentados a azul-claro e os eventos e sinais de saída são apresentados a verde.

Cada objecto pode ter uma ou mais inscrições, incluindo nomes de identificação e comentários. Os nomes dos lugares são apresentados a vermelho e os nomes das transições a verde. Os identificadores dos sinais e eventos estão a rosa. Expressões de guarda das transições são apresentadas a cinzento. Expressões de acções de saída de lugares e de transições estão a azul.

¹ modelo INDIN07_park1in1out.pnml disponível em [16]

Os triângulos (eventos) apontam para cima ou para baixo consoante a direcção do flanco (*edge*) do evento, traduzindo-se respectivamente em eventos ascendentes (eventos *ArriveIn* e *Identified*) ou descendentes (evento *ArriveOut*).

Na RdP, eventos associados a transições estão representados por pequenas setas nas laterais das transições: eventos de entrada à esquerda da transição (*ArriveIn*, *Identified* e *ArriveOut*) e eventos de saída à direita da transição (neste caso não existem).

Expressões de guarda e de acções de saída estão apresentadas por baixo do objecto correspondente.

Neste exemplo concreto, apenas existem os eventos de entrada *ArriveIn* e *Identified* e *ArriveOut*. Os sinais de entrada são: *GotTicket*, *arrive*, *pay*, *leave*. E os sinais de saída são: *GateInOpen*, *GateOutOpen*.

O parque terá a lotação para 100 lugares de estacionamento. Os lugares *FreePlaces* e *CarInsideZone* representam, respectivamente, o número de lugares de estacionamento livres e ocupados no parque. Assim, inicialmente, *FreePlaces* será 100 e *CarInsideZone* será 0.

Assim, antes da Simulação do modelo, através do Wave4IOPT podemos observar estado do sistema com a sua marcação inicial, tal como, os eventos e sinais de entrada.

O estado inicial da RdP é exportado do Simulador IOPT para a ferramenta Wave4IOPT a partir de um ficheiro *JSON1In1Out_beforeSim*. Pelo facto do seu conteúdo ser muito extenso, optou-se por apresentar o ficheiro completo no **Anexo I.1** e somente uma parte do ficheiro seguidamente:

```
{
  "name": "p_3012/FreePlaces",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 100,
  "behaviourVector": [100],
```

```
    "viewMode": "simpleComp",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "p_3017/CarInsideZone",
    "type": "internal",
    "subtype": "place",
    "minValue": 0,
    "maxValue": 4,
    "behaviourVector": [0],
    "viewMode": "simpleComp",
    "numericalBase": "d",
    "visibility": 1
  },
}
```

Assim, antes da simulação da rede e após leitura do ficheiro JSON (*JSON1In1Out_beforeSim*), este será o comportamento dos lugares e dos eventos e sinais de entrada no Wave4IOPT:

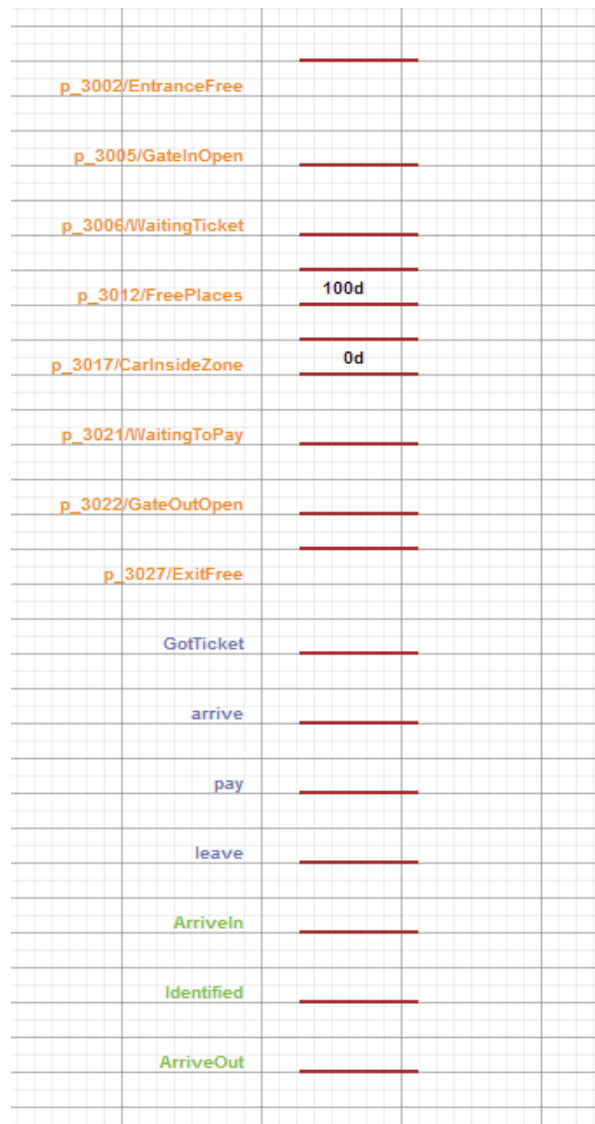


Figura 4.2: Representação no Wave4IOPT do estado inicial da RdP antes da simulação do parque de estacionamento com 1 entrada e 1 saída

Por omissão, no WaveIOPT, o nome de todos os lugares são apresentados a laranja, dos sinais a azul e dos eventos a verde. Os nomes dos lugares surgem representados a partir da concatenação levada a cabo pelo Simulador ao juntar os seus *id* e nome na rede.

Estão somente representadas as formas de onda dos lugares e dos eventos e sinais de entrada, pelo facto de serem os únicos cujo valor se pode alterar no

Wave4IOPT. Deste modo, caso se pretenda, poder-se-á alterar os valores do *step* inicial da RdP.

Após as edições que se pretendam efectuar, efectua-se a gravação do ficheiro JSON que será posteriormente lido pelo Simulador.

Seguidamente, será efectuada a simulação pelo Simulador IOPT de modo a testar a evolução da rede, ou seja, a dinâmica do parque de estacionamento.

Deste modo, estando a entrada do parque livre (lugar *EntranceFree*) e caso um veículo chegue à entrada, o sinal de entrada *arrive* fica com o valor 1 gerando-se um evento ascendente *Arrive In* e a transição *entrance_occupied* é disparada, significando que um veículo chegou à entrada do parque de estacionamento. O lugar *EntranceFree* fica sem a marca que antes tinha e o lugar *WaitingTicket* fica com uma marca, indicando que um veículo está na entrada aguardando por um *ticket* para entrar no parque. O condutor recebe o ticket quando o sinal *GotTicket* fica 1, existindo um evento ascendente *Identified* sendo disparada a transição *got_ticket*. Assim, o lugar *GateInOpen* passa a ter uma marca significando que é aberta a cancela de entrada. Após o sinal *arrive* ficar a 0, é criado um evento descendente chamado *ArriveOut* associado à transição *entrance_empty*. Esta transição é disparada significando que o carro já não está na entrada e que a cancela será fechada gerando-se um sinal de saída *GateInOpen*. O lugar *GateInOpen* fica sem marca e o lugar *EntranceFree* fica com uma marca, significando que a entrada ficou livre e disponível para outro veículo.

Quanto à dinâmica da rede para a saída de veículos do parque de estacionamento, o funcionamento é idêntico.

Quando a saída está livre o lugar P6 está marcado. Quando um carro se apresenta junto à saída e é colocado no sinal *leave* a 1, ficando-se à espera que o pagamento foi realizado (lugar *WaitingToPay*). Após o sinal *pay* estar a 1, indicativo do pagamento estar realizado, a cancela de saída é aberta através do sinal de saída

GateOutOpen, sendo libertado um lugar de estacionamento. Quando o sinal de entrada *leave* está com valor 0, significa que o veículo já não se encontra junto à cancela de saída sendo fechada, ficando assim a saída à espera de um novo condutor que vá sair do parque de estacionamento.

Assim, após a simulação, é exportado o histórico dessa simulação a partir de um ficheiro *JSON1In1Out_afterSim*. A partir da funcionalidade de abertura de ficheiros JSON, a ferramenta Wave4IOPT apresenta o histórico da evolução da RdP do Simulador IOPT. Pelo facto do seu conteúdo ser muito extenso, optou-se por apresentar o ficheiro completo no **Anexo I.2** e somente uma parte do ficheiro seguidamente:

```
{
  "name": "p_3012/FreePlaces",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 100,
  "behaviourVector":
[100,100,100,99,99,99,99,98,98,98,98,98,97,97,97,97,96,96,96,96,97,97,97,97
,97,97,97,97,97,97,98,98,98,98,97,98,98,98,98,98,98],
  "viewMode": "simpleComp",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "p_3017/CarInsideZone",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 4,
  "behaviourVector":
[0,0,0,1,1,1,1,2,2,2,2,2,3,3,3,3,4,4,4,4,3,3,3,3,3,3,3,3,3,3,2,2,2,2,3,2,2,
2,2,2,2],
  "viewMode": "simpleComp",
  "numericalBase": "d",
  "visibility": 1
},
}
```

Após o Wave4IOPT ler o ficheiro JSON (*JSON1In1Out_afterSim*), este foi o comportamento observável dos lugares e dos eventos e sinais de entrada:

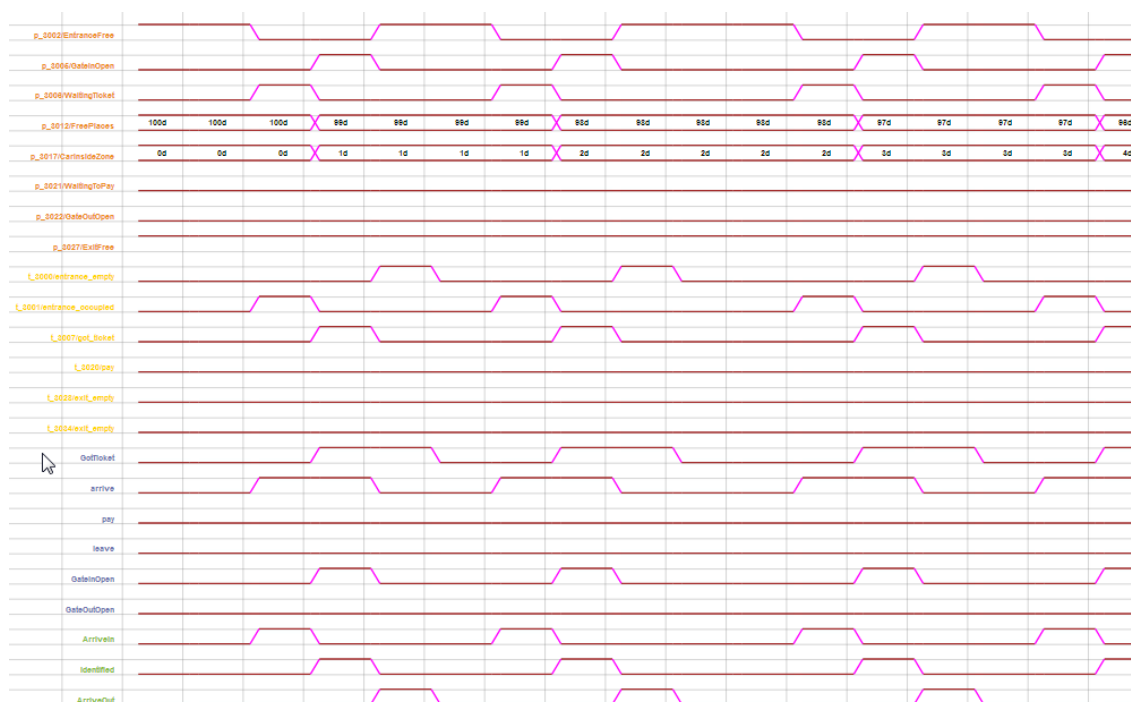


Figura 4.3: Aspecto geral no Wave4IOPT após a simulação do parque de estacionamento com 1 entrada e 1 saída.

A partir da funcionalidade de ampliação pode-se ver mais em pormenor a evolução da Forma de Onda que se pretende.

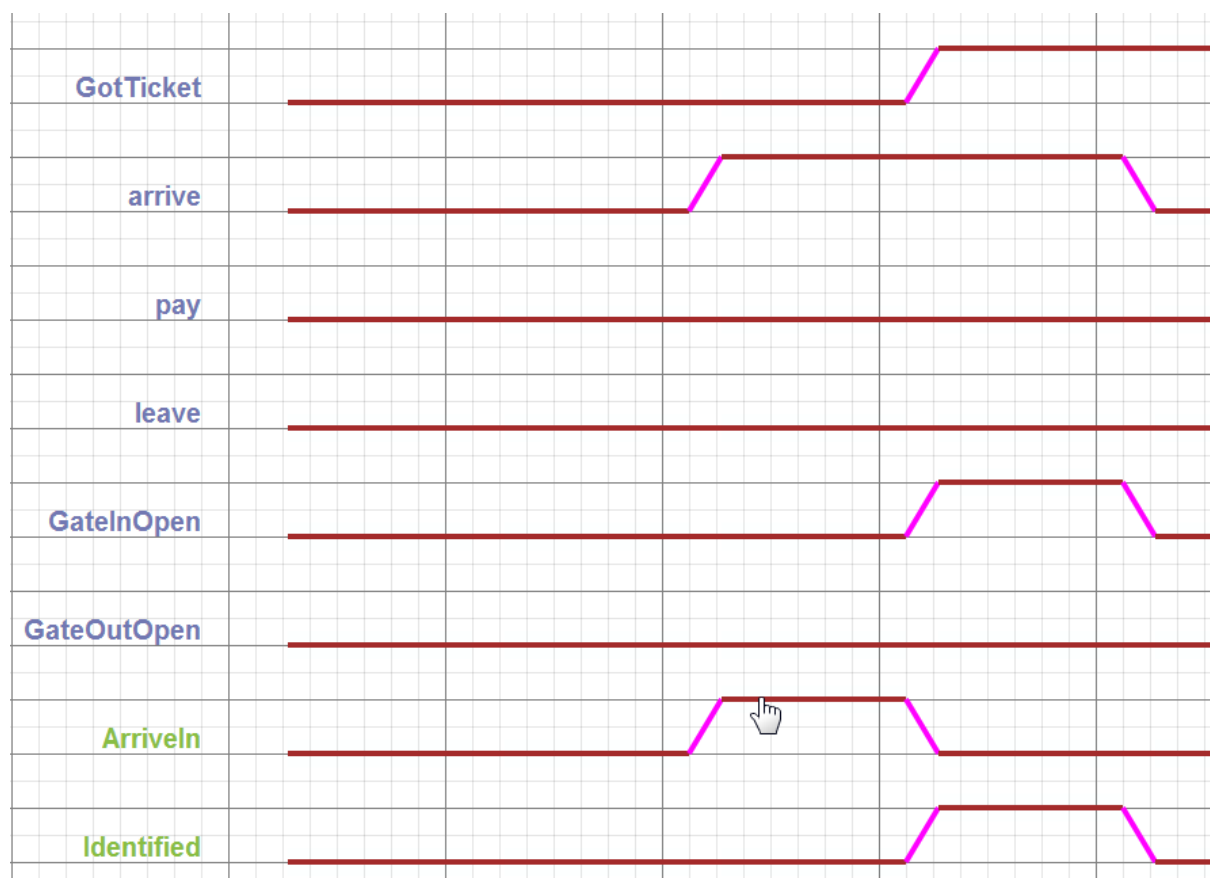


Figura 4.4: Representação com zoom a 150% no Wave4IOPT da RdP após a simulação do parque de estacionamento com 1 entrada e 1 saída.

Analisando-se as diversas Formas de Onda, comprova-se que estas traduzem fielmente a evolução da RdP proveniente da simulação.

A partir daqui, poderemos dar uso a qualquer uma das funcionalidades de visualização e edição do Wave4IOPT, descritas em detalhe no **Capítulo 3** desta dissertação. Assim, efectuaram-se diversos testes de forma a validar o funcionamento de todas as funcionalidades da ferramenta, tendo-se verificado o seu correcto funcionamento.

Com este sub-capítulo demonstrou-se através de um exemplo de um parque de estacionamento com uma entrada e uma saída, a aplicabilidade da ferramenta desenvolvida no âmbito deste trabalho.

4.2. Parque de estacionamento com duas entradas e uma saída

De forma análoga ao ponto anterior, a partir de um exemplo de uma RdP IOPT disponível na IOPT-Tools, pretende-se modelar um controlador de entradas e saídas de um parque de estacionamento com duas entradas e uma saída, com capacidade para 100 veículos (**Figura 4.5**).

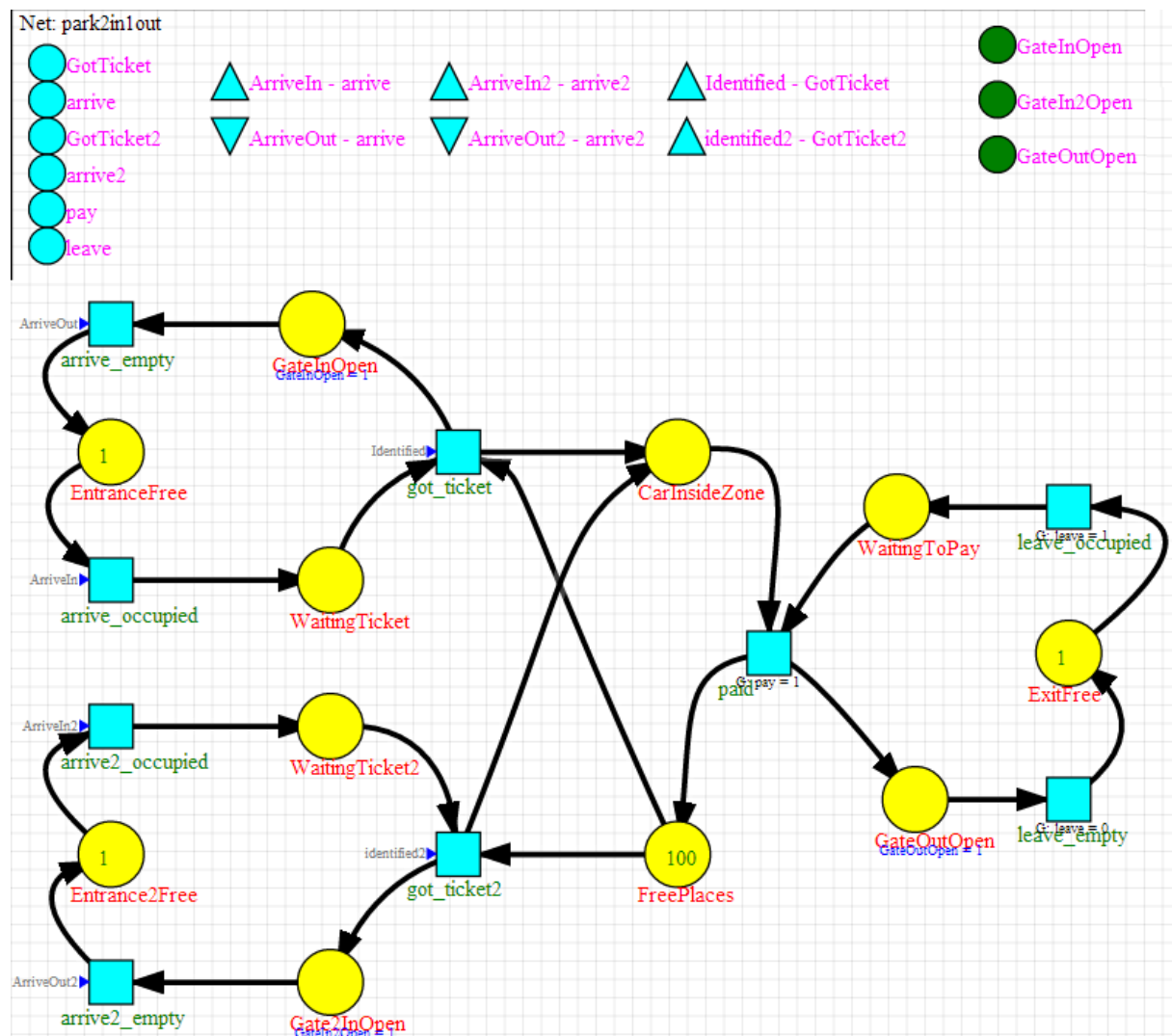


Figura 4.5: RdP IOPT de um controlador de entradas e saídas de um parque de estacionamento com 2 entradas e 1 saída.¹

¹ modelo park2in1out.pnml disponível em [16]

Comparando com o exemplo do ponto anterior, este parque de estacionamento será muito semelhante. No entanto, terá adicionalmente uma segunda entrada no parque, representada no modelo pelos lugares *Gate2InOpen*, *Entrance2Free* e *WaitingTicket2* e pelas transições *arrive2_empty*, *arrive2_occupied* e *got_ticket2*.

Observa-se, igualmente, que existem os seguintes eventos de entrada: *ArriveIn* e *Identified*, *ArriveOut*, *ArriveIn2* e *identified2* e *ArriveOut2*. Quanto aos sinais de entrada: *GotTicket*, *arrive*, *pay*, *leave*, *GotTicket2* e *arrive2*. E os sinais de saída são: *GateInOpen*, *GateOutOpen* e *GateIn2Open*.

Assim, antes da Simulação do modelo, através do Wave4IOPT podemos observar estado do sistema com a sua marcação inicial, bem como, os eventos e sinais de entrada.

O estado inicial da RdP é exportado do Simulador IOPT para a ferramenta Wave4IOPT a partir de um ficheiro *JSON2In1Out_beforeSim*. Pelo facto do seu conteúdo ser muito extenso, optou-se por apresentar o ficheiro completo no **Anexo I.3** e somente uma parte do ficheiro seguidamente:

```
{
  "name": "identified2",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "ArriveIn2",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
```

Assim, antes da simulação da rede e após leitura do ficheiro JSON (*JSON2In1Out_beforeSim*), este será o comportamento dos lugares e dos eventos e sinais de entrada no Wave4IOPT:

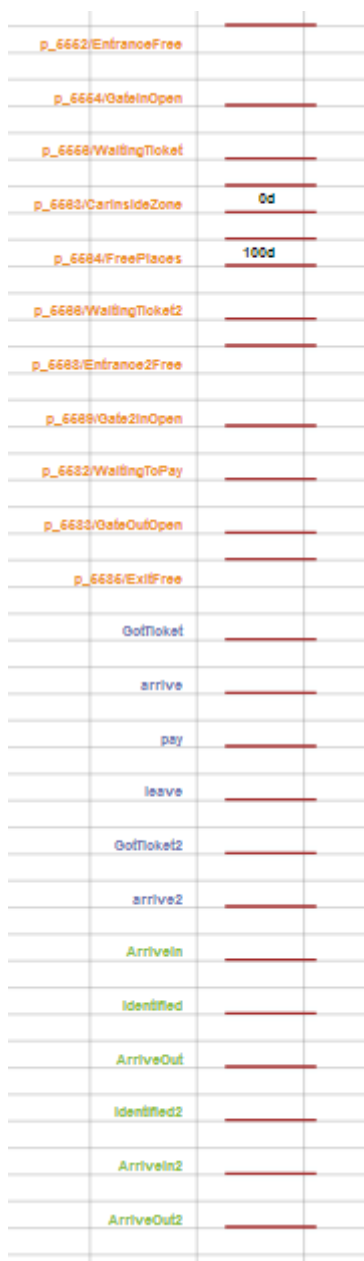


Figura 4.6: Representação no Wave4IOPT do estado inicial da RdP antes da simulação do parque de estacionamento com 2 entradas e 1 saída.

Nesta fase, poder-se-ia alterar os valores da marcação inicial dos lugares, bem como dos eventos e sinais de entrada, mostrados na **Figura 4.6**.

No caso de se ter editado algum valor, procede-se à gravação efectua-se a gravação desse ficheiro JSON que será posteriormente lido pelo Simulador.

Seguidamente, será efectuada a simulação pelo Simulador IOPT de modo a testar a evolução da RdP.

Assim, quanto à evolução da RdP, no caso de não existir qualquer pedido de entrada de veículo pela segunda entrada, a primeira entrada é idêntica à do parque de estacionamento com uma entrada e uma saída, como descrito no ponto anterior. No que toca à segunda entrada, o comportamento é muito semelhante à primeira entrada com uma situação diferente: no caso estarem 99 lugares ocupados no parque de estacionamento e existir um pedido de entrada nas duas entradas no mesmo instante (as transições *got_ticket* e *got_ticket2* estarem ambas habilitadas e dispararem no mesmo passo de execução) existirá uma situação de conflito. Assim, tendo em conta que é possível associar-se uma prioridade para as transições nas RdP IOPT, foi estabelecido que a transição *got_ticket* teria prioridade sobre a transição *got_ticket2*. Desta forma, nesta situação, somente a transição *got_ticket* seria disparada, ficando o lugar *WaitingTicket2* “à espera” que houvesse pelo menos um lugar disponível no parque de estacionamento.

Deste modo, após a realização de uma simulação, foi exportado o histórico dessa simulação a partir do ficheiro *JSON2In1Out_afterSim*. Pelo facto do seu conteúdo ser muito extenso, optou-se por apresentar o ficheiro completo no **Anexo I.4** e somente uma parte do ficheiro seguidamente:

```
{
  "name": "identified2",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
```

```

    "behaviourVector": [0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "ArriveIn2",
    "type": "input",
    "subtype": "event",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
},

```

O Wave4IOPT abre esse ficheiro JSON, podendo-se finalmente observar o histórico da evolução da RdP do Simulador IOPT para o modelo de um parque de estacionamento com duas entradas e uma saída:

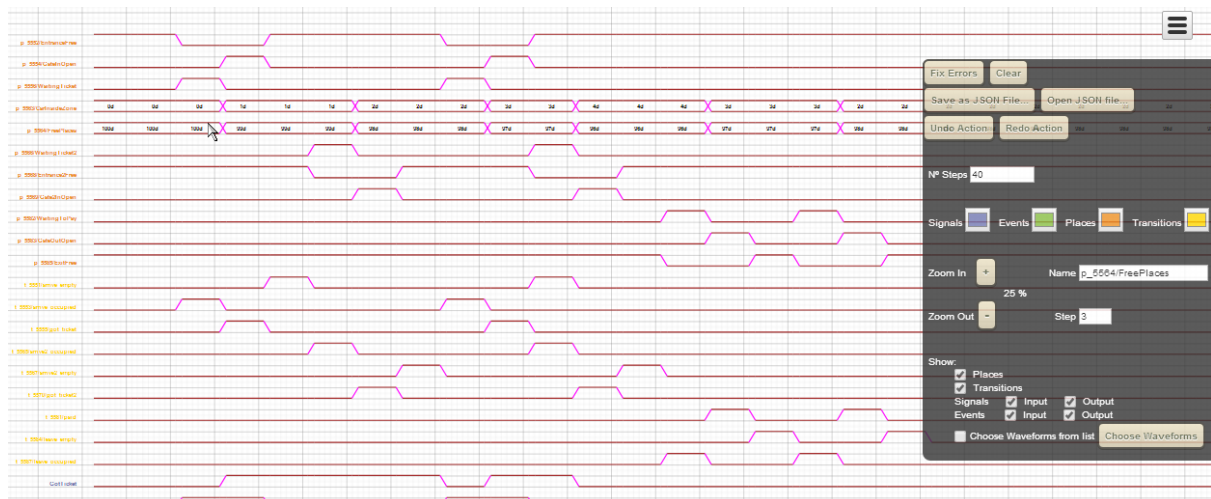


Figura 4.7: Aspecto geral no Wave4IOPT após a simulação do parque de estacionamento com 2 entradas e 1 saída.

Utilizando a funcionalidade de ampliação pode-se ver mais em pormenor a evolução da Forma de Onda que se pretender.

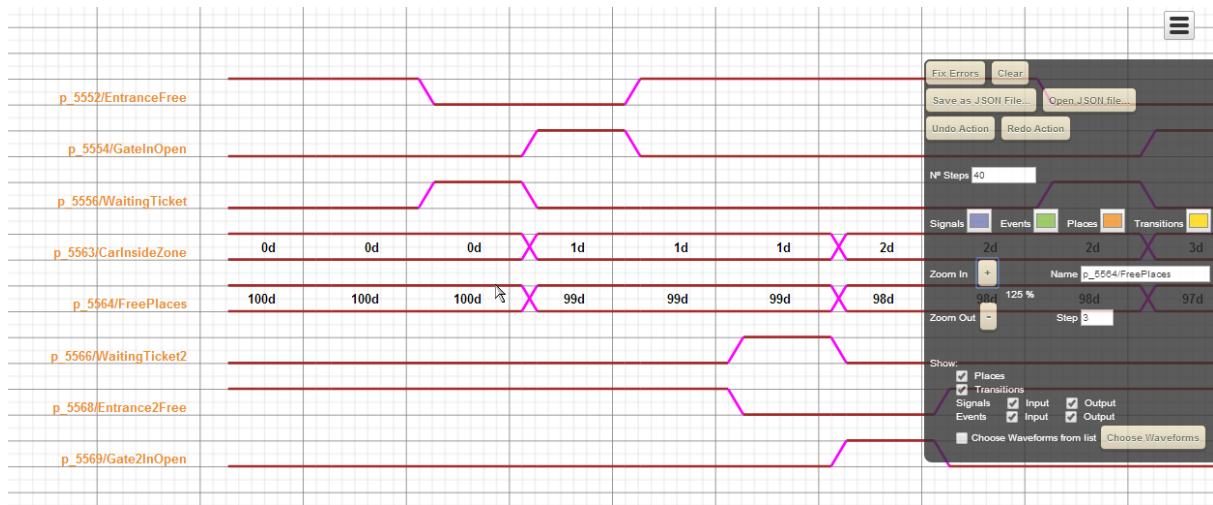


Figura 4.8: Representação com zoom a 125% no Wave4IOPT da RdP após a simulação do parque de estacionamento com 2 entradas e 1 saída.

Analisando-se as diversas Formas de Onda, comprova-se que estas traduzem fielmente a evolução da RdP proveniente da simulação.

Após a sua análise, realizaram-se alguns testes de modo a tirar partido das funcionalidades de edição e visualização da ferramenta desenvolvida, não existindo nada de anómalo a reportar.

Neste sub-capítulo, devido à semelhança deste modelo com o do sub-capítulo anterior, pretendeu-se dar ênfase ao que era diferente, nomeadamente: aos conflitos e à sua resolução, através de atribuição de prioridades às transições. Deste modo, através de um exemplo de um parque de estacionamento um pouco mais complexo que o anterior (com duas entradas e uma saída), demonstrou-se a aplicabilidade do Wave4IOPT neste trabalho.



Conclusões

Nesta dissertação desenvolveu-se um Editor e Visualizador de Formas de Onda, acessível a partir de um browser, com aplicação a controladores digitais especificados com modelos de Redes de Petri IOPT.

Esta ferramenta aproveita as vantagens de tecnologias como o SVG, de forma a oferecer uma ferramenta que não permite a perda de qualidade na visualização das formas de onda (de sinais e eventos de entrada e saída), resultado do histórico de uma simulação de uma RdP IOPT.

A portabilidade desta ferramenta para outros tipos de sinais digitais é um aspecto em ter em conta, na medida em que poderá oferecer funcionalidades de edição, visualização e análise, desde que sejam preenchidos os requisitos da estrutura do ficheiro JSON, a ser lido pelo Wave4IOPT.

Tendo este último ponto em mente, seria interessante, futuramente, o Wave4IOPT dar suporte a outros formalismos que modelem sinais discretos, como por exemplo, Estadogramas (*Statecharts*).

No que toca a trabalho futuro, poder-se-iam adicionar mais funcionalidades interessantes ao Wave4IOPT, nomeadamente:

- botões *import/export* – garantir a comunicação directa entre as ferramentas Wave4IOPT e o Simulador IOPT, através de funcionalidades de importação / exportação, de forma a que deixe de ser uma necessidade a existência de um ficheiro JSON como intermediário.
- um botão de ajuda (*help*) – que dê acesso a um manual ou tutorial em vídeo, de forma a que o utilizador possa tirar possíveis dúvidas e tire partido de todas as funcionalidades da ferramenta.
- um marcador – ter uma caixa de opção para uso ou não de um marcador, representado por uma linha vertical, que passe por todas as Formas de Onda. O intuito será guiar o utilizador quando estão representadas muitas Formas de Onda.
- *clock* – ter uma caixa de opção com a possibilidade de se poder visualizar ou não o sinal de *clock*.
- abrir ficheiros noutros formatos – dar suporte à abertura de históricos de simulações que estejam em ficheiros noutros formatos para além de JSON.

Num futuro breve, a integração desta ferramenta no ambiente de ferramentas IOPT-Tools, será certamente uma mais-valia para todos aqueles que trabalham na modelação de controladores digitais e em particular para o projecto “Ambiente de desenvolvimento de sistemas embutidos baseado em redes de Petri” – Petri-Rig.

Referências Bibliográficas

- [1] Craig Larman, “Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development”, 3rd Edition, 2005.
- [2] J. P. Barros, “Casos de Uso e Respectivos Diagramas”, Escola Superior de Tecnologia e Gestão Instituto Politécnico de Beja, 2009.
- [3] Carl Adam Petri, “Kommunikation mit Automaten”, Schriften des IIM No 2, Institut für Instrumentelle Mathematik, Bonn, 1962. Traduzida para inglês como: Communication with Automata, Technical Report RADC-TR-65-377, Griffiths Air Force Base, New York, Vol. 1, Suppl. 1, 1966.
- [4] J. L. Peterson, “Petri Net Theory and the Modeling of Systems”, Prentice Hall, Inc., 190 pág., Junho de 1981.
- [5] Murata, Tadao, “Petri Nets: Properties Analysis and Applications”, Proceedings of the IEEE, Vol. 77, No. 4, April 1989.
- [6] L. F. dos Santos Gomes, “Redes de Petri Reactivas e Hierárquicas – integração de formalismos no projecto de sistemas reactivos de tempo-real”, Tese de Doutoramento, Faculdade de Ciências e Tecnologia – Universidade Nova de Lisboa, 1997.
- [7] R. Nunes, L. Gomes, J.P. Barros, “A graphical editor for the input-output place-transition petri net class”, Emerging Technologies and Factory Automation, 2007, Universidade Nova de Lisboa.
- [8] Reisig, Wolfgang, “Petri nets: an introduction”, Springer-Verlag New York Inc., 1985.
- [9] L. Gomes, F. Moutinho, F. Pereira, “IOPT-TOOLS – A Web based tool framework for embedded systems controller development using Petri nets”, Field Programmable Logic and Applications (FPL), 23rd International Conference on, 2013.

- [10] M. Weber, E. Kindler, “The Petri Net Markup Language”, HU Berlin: Institut für Informatik, 2003.
- [11] Luís Abreu, “JavaScript”, 2ª Edição. Colaboração de João Paulo Carreiro, 2013.
- [12] Standard ECMA – 262, 6th Edition/June2015 ECMAScript 2015 Language Specification, 545 pág..
- [13] J. David Eisenberg, Amelia Bellamy-Royds, “SVG Essentials – Producing Scalable Vector Graphics with XML”, 2nd Edition”, 366 pág., 2014.

Referências Electrónicas

Acedidas em Setembro de 2015:

- [14] *World of Petri Nets*:

<http://www.informatik.uni-hamburg.de/TGI/PetriNets/>

- [15] *EPWave*:

<http://www.victoreda.com>

- [16] The IOPT-Tools Website:

<http://gres.uninova.pt>

- [17] <http://json.org>

- [18] *Scalable Vector Graphics (SVG)*:

<http://www.w3.org/Graphics/SVG/>

- [19] <http://jsonviewer.stack.hu>

Anexo I

I.1 Ficheiro JSON1In1Out_beforeSim

```
[  
  {  
    "name": "p_3002/EntranceFree",  
    "type": "internal",  
    "subtype": "place",  
    "minValue": 0,  
    "maxValue": 1,  
    "behaviourVector": [1],  
    "viewMode": "normal",  
    "numericalBase": "d",  
    "visibility": 1  
  },  
  {  
    "name": "p_3005/GateInOpen",  
    "type": "internal",  
    "subtype": "place",  
    "minValue": 0,  
    "maxValue": 1,  

```

```
"behaviourVector": [0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "p_3006/WaitingTicket",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "p_3012/FreePlaces",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 100,
  "behaviourVector": [100],
  "viewMode": "simpleComp",
  "numericalBase": "d",
  "visibility": 1
```

```
},  
{  
  "name": "p_3017/CarInsideZone",  
  "type": "internal",  
  "subtype": "place",  
  "minValue": 0,  
  "maxValue": 1,  
  "behaviourVector": [0],  
  "viewMode": "normal",  
  "numericalBase": "d",  
  "visibility": 1  
},  
{  
  "name": "p_3021/WaitingToPay",  
  "type": "internal",  
  "subtype": "place",  
  "minValue": 0,  
  "maxValue": 1,  
  "behaviourVector": [0],  
  "viewMode": "normal",  
  "numericalBase": "d",  
  "visibility": 1  
},  
{  
  "name": "p_3022/GateOutOpen",  
  "type": "internal",
```

```
"subtype": "place",
"minValue": 0,
"maxValue": 1,
"behaviourVector": [0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "p_3027/ExitFree",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [1],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_3000/entrance_empty",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
```



```
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "t_3001/entrance_occupied",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_3007/got_ticket",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
```

```
{
  "name": "t_3020/pay",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_3028/exit_empty",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_3034/exit_empty",
  "type": "internal",
  "subtype": "transition",
```

```
"minValue": 0,  
"maxValue": 1,  
"behaviourVector": [0],  
"viewMode": "normal",  
"numericalBase": "d",  
"visibility": 1  
},  
{  
  "name": "GotTicket",  
  "type": "input",  
  "subtype": "signal",  
  "minValue": 0,  
  "maxValue": 1,  
  "behaviourVector": [0],  
  "viewMode": "normal",  
  "numericalBase": "d",  
  "visibility": 1  
},  
{  
  "name": "arrive",  
  "type": "input",  
  "subtype": "signal",  
  "minValue": 0,  
  "maxValue": 1,  
  "behaviourVector": [0],  
  "viewMode": "normal",
```

```
"numericalBase": "d",
"visibility": 1
},
{
  "name": "pay",
  "type": "input",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "leave",
  "type": "input",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
```

```
"name": "GateInOpen",
"type": "output",
"subtype": "signal",
"minValue": 0,
"maxValue": 1,
"behaviourVector": [0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "GateOutOpen",
  "type": "output",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "ArriveIn",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
```

```
"maxValue": 1,
"behaviourVector": [0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "Identified",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "ArriveOut",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
```

```

    "visibility": 1
  }
]

```

I.2 Ficheiro JSON1In1Out_afterSim

```

[
  {
    "name": "p_3002/EntranceFree",
    "type": "internal",
    "subtype": "place",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector":
[1,1,0,0,1,1,0,0,1,1,1,0,0,1,1,0,0,1,1,1,1,1,1,0,0,1,1,1,1,1,1,0,0,1,1,1,1,1],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "p_3005/GateInOpen",

```

```
"type": "internal",
"subtype": "place",
"minValue": 0,
"maxValue": 1,
"behaviourVector":
[0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
"name": "p_3006/WaitingTicket",
"type": "internal",
"subtype": "place",
"minValue": 0,
"maxValue": 1,
"behaviourVector":
[0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
"name": "p_3012/FreePlaces",
"type": "internal",
"subtype": "place",
"minValue": 0,
```



```

    "maxValue": 100,
    "behaviourVector":
[100,100,100,99,99,99,99,98,98,98,98,98,97,97,97,97,96,96,96,96,97,97,97,97,97,97,9
7,97,98,98,98,98,97,98,98,98,98,98],
    "viewMode": "simpleComp",
    "numericalBase": "d",
    "visibility": 1
},
{
    "name": "p_3017/CarInsideZone",
    "type": "internal",
    "subtype": "place",
    "minValue": 0,
    "maxValue": 4,
    "behaviourVector":
[0,0,0,1,1,1,1,2,2,2,2,2,3,3,3,3,4,4,4,4,3,3,3,3,3,3,3,3,2,2,2,2,3,2,2,2,2,2,2],
    "viewMode": "simpleComp",
    "numericalBase": "d",
    "visibility": 1
},
{
    "name": "p_3021/WaitingToPay",
    "type": "internal",
    "subtype": "place",
    "minValue": 0,
    "maxValue": 1,

```



```

    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "t_3000/entrance_empty",
    "type": "internal",
    "subtype": "transition",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector":
[0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "t_3001/entrance_occupied",
    "type": "internal",
    "subtype": "transition",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector":
[0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },

```

```
{
  "name": "t_3007/got_ticket",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector":
[0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_3020/pay",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_3028/exit_empty",
  "type": "internal",
```

```
{
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_3034/exit_empty",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "GotTicket",
  "type": "input",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
```

```
"behaviourVector":
[0,0,0,1,1,0,0,1,1,0,0,0,1,1,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "arrive",
  "type": "input",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector":
[0,0,1,1,0,0,1,1,0,0,0,1,1,0,0,1,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "pay",
  "type": "input",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector":
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,1,0,0,0,0],
  "viewMode": "normal",
```

```
"numericalBase": "d",  
  "visibility": 1  
},  
  
{  
  "name": "leave",  
  "type": "input",  
  "subtype": "signal",  
  "minValue": 0,  
  "maxValue": 1,  
  "behaviourVector":  
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,0,0,0,0,0],  
  "viewMode": "normal",  
  "numericalBase": "d",  
  "visibility": 1  
},  
  
{  
  "name": "GateInOpen",  
  "type": "output",  
  "subtype": "signal",  
  "minValue": 0,  
  "maxValue": 1,  
  "behaviourVector":  
[0,0,0,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0],  
  "viewMode": "normal",  
  "numericalBase": "d",  
  "visibility": 1  
},
```



```

    "subtype": "event",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector":
[0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
},
{
    "name": "ArriveOut",
    "type": "input",
    "subtype": "event",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector":
[0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
}
]

```

I.3 Ficheiro JSON2In1Out_beforeSim

```
[
  {
    "name": "p_5552/EntranceFree",
    "type": "internal",
    "subtype": "place",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [1],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "p_5554/GateInOpen",
    "type": "internal",
    "subtype": "place",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
```

```
"name": "p_5556/WaitingTicket",
"type": "internal",
"subtype": "place",
"minValue": 0,
"maxValue": 1,
"behaviourVector": [0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "p_5563/CarInsideZone",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "p_5564/FreePlaces",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
```

```
"maxValue": 100,
"behaviourVector": [100],
"viewMode": "simpleComp",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "p_5566/WaitingTicket2",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "p_5568/Entrance2Free",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [1],
  "viewMode": "normal",
  "numericalBase": "d",
```

```
"visibility": 1
},
{
  "name": "p_5569/Gate2InOpen",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "p_5582/WaitingToPay",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "p_5583/GateOutOpen",
```

```
"type": "internal",
"subtype": "place",
"minValue": 0,
"maxValue": 1,
"behaviourVector": [0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "p_5585/ExitFree",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [1],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_5551/arrive_empty",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
```

```
"behaviourVector": [0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "t_5553/arrive_occupied",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_5555/got_ticket",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
```

```
},
{
  "name": "t_5565/arrive2_occupied",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_5567/arrive2_empty",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_5570/got_ticket2",
  "type": "internal",
```



```
"subtype": "transition",
"minValue": 0,
"maxValue": 1,
"behaviourVector": [0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "t_5581/paid",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_5584/leave_empty",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
```

```
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "t_5587/leave_occupied",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "GotTicket",
  "type": "input",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
```

```
{
  "name": "arrive",
  "type": "input",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "pay",
  "type": "input",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "leave",
  "type": "input",
  "subtype": "signal",
```

```
"minValue": 0,  
"maxValue": 1,  
"behaviourVector": [0],  
"viewMode": "normal",  
"numericalBase": "d",  
"visibility": 1  
},  
{  
  "name": "GotTicket2",  
  "type": "input",  
  "subtype": "signal",  
  "minValue": 0,  
  "maxValue": 1,  
  "behaviourVector": [0],  
  "viewMode": "normal",  
  "numericalBase": "d",  
  "visibility": 1  
},  
{  
  "name": "arrive2",  
  "type": "input",  
  "subtype": "signal",  
  "minValue": 0,  
  "maxValue": 1,  
  "behaviourVector": [0],  
  "viewMode": "normal",
```

```
"numericalBase": "d",
"visibility": 1
},
{
  "name": "GateInOpen",
  "type": "output",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "GateOutOpen",
  "type": "output",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
```

```
"name": "GateIn2Open",
"type": "output",
"subtype": "signal",
"minValue": 0,
"maxValue": 1,
"behaviourVector": [0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "ArriveIn",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "Identified",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
```

```
"maxValue": 1,  
"behaviourVector": [0],  
"viewMode": "normal",  
"numericalBase": "d",  
"visibility": 1  
},  
{  
  "name": "ArriveOut",  
  "type": "input",  
  "subtype": "event",  
  "minValue": 0,  
  "maxValue": 1,  
  "behaviourVector": [0],  
  "viewMode": "normal",  
  "numericalBase": "d",  
  "visibility": 1  
},  
{  
  "name": "identified2",  
  "type": "input",  
  "subtype": "event",  
  "minValue": 0,  
  "maxValue": 1,  
  "behaviourVector": [0],  
  "viewMode": "normal",  
  "numericalBase": "d",
```

```
"visibility": 1
},
{
  "name": "ArriveIn2",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "ArriveOut2",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
}
]
```


I.4 Ficheiro JSON12n1Out_afterSim

```
[
  {
    "name": "p_5552/EntranceFree",
    "type": "internal",
    "subtype": "place",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [1,1,0,0,1,1,1,0,0,1,1,1,1,1,1,1],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "p_5554/GateInOpen",
    "type": "internal",
    "subtype": "place",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
```

```
"name": "p_5556/WaitingTicket",
"type": "internal",
"subtype": "place",
"minValue": 0,
"maxValue": 1,
"behaviourVector": [0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "p_5563/CarInsideZone",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 4,
  "behaviourVector": [0,0,0,1,1,1,2,2,2,3,3,4,4,4,3,3,3,2,2,2],
  "viewMode": "simpleComp",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "p_5564/FreePlaces",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
```

```
"maxValue": 100,
"behaviourVector": [100,100,100,99,99,99,98,98,98,97,97,96,96,96,97,97,97,98,98,98],
"viewMode": "simpleComp",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "p_5566/WaitingTicket2",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "p_5568/Entrance2Free",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [1,1,1,1,1,0,0,1,1,1,0,0,1,1,1,1,1,1,1,1],
  "viewMode": "normal",
  "numericalBase": "d",
```

```
"visibility": 1
},
{
  "name": "p_5569/Gate2InOpen",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "p_5582/WaitingToPay",
  "type": "internal",
  "subtype": "place",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "p_5583/GateOutOpen",
```

```

    "type": "internal",
    "subtype": "place",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "p_5585/ExitFree",
    "type": "internal",
    "subtype": "place",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [1,1,1,1,1,1,1,1,1,1,1,1,0,0,1,0,0,1,1],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "t_5551/arrive_empty",
    "type": "internal",
    "subtype": "transition",
    "minValue": 0,
    "maxValue": 1,

```

```
"behaviourVector": [0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "t_5553/arrive_occupied",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_5555/got_ticket",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
}
```

```
,
{
  "name": "t_5565/arrive2_occupied",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_5567/arrive2_empty",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_5570/got_ticket2",
  "type": "internal",
```

```
"subtype": "transition",
"minValue": 0,
"maxValue": 1,
"behaviourVector": [0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "t_5581/paid",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "t_5584/leave_empty",
  "type": "internal",
  "subtype": "transition",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0],
```



```

    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "t_5587/leave_occupied",
    "type": "internal",
    "subtype": "transition",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "GotTicket",
    "type": "input",
    "subtype": "signal",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [0,0,0,1,1,1,1,1,0,1,1,0,0,0,0,0,0,0,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },

```

```
{
  "name": "arrive",
  "type": "input",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,1,1,0,0,0,0,1,1,0,0,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "pay",
  "type": "input",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,1,1,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "leave",
  "type": "input",
  "subtype": "signal",
```

```

    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,1,1,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "GotTicket2",
    "type": "input",
    "subtype": "signal",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [0,0,0,0,0,0,1,1,0,0,0,1,0,0,0,0,0,0,0,0],
    "viewMode": "normal",
    "numericalBase": "d",
    "visibility": 1
  },
  {
    "name": "arrive2",
    "type": "input",
    "subtype": "signal",
    "minValue": 0,
    "maxValue": 1,
    "behaviourVector": [0,0,0,0,0,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0],
    "viewMode": "normal",

```

```
"numericalBase": "d",
"visibility": 1
},
{
  "name": "GateInOpen",
  "type": "output",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "GateOutOpen",
  "type": "output",
  "subtype": "signal",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
```

```
"name": "GateIn2Open",
"type": "output",
"subtype": "signal",
"minValue": 0,
"maxValue": 1,
"behaviourVector": [0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "ArriveIn",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "Identified",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
```

```
"maxValue": 1,
"behaviourVector": [0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0],
"viewMode": "normal",
"numericalBase": "d",
"visibility": 1
},
{
  "name": "ArriveOut",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "identified2",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
```

```
"visibility": 1
},
{
  "name": "ArriveIn2",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
},
{
  "name": "ArriveOut2",
  "type": "input",
  "subtype": "event",
  "minValue": 0,
  "maxValue": 1,
  "behaviourVector": [0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0],
  "viewMode": "normal",
  "numericalBase": "d",
  "visibility": 1
}
]
```